



Open Source Used In talos_cua-clouduri-engine-go main-git-0fc5a72-26

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-2482142597

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this [form](#).

In your requests please include the following reference number 78EE117C99-2482142597

Contents

1.1 lz4 1.3.1

1.1.1 Available under license

1.2 errwrap 1.1.0

1.2.1 Available under license

1.3 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.3.1 Available under license

1.4 kelseyhightower-envconfig 1.4.0

1.4.1 Available under license

1.5 hcl 1.0.0

1.5.1 Available under license

1.6 open-telemetry-opentelemetry-collector-contrib 0.61.0

1.6.1 Available under license

1.7 githubcommitchellhmapstructure 1.5.0

1.7.1 Available under license

1.8 httpsnoop 1.0.4

1.8.1 Available under license

1.9 libmd 1.1.0-r0

1.9.1 Available under license

1.10 zlib 1.3.1

1.10.1 Available under license

1.11 google-uuid 1.6.0

1.11.1 Available under license

1.12 libbsd 0.12.2-r0

1.12.1 Available under license

1.13 xxhash 2.3.0

1.13.1 Available under license

1.14 githubcomhashicorp/hcl 1.0.0

1.14.1 Available under license

1.15 github.com/hashicorp/go-secure-stdlib/strutil 0.1.2

1.15.1 Available under license

1.16 goglob 1.0.0

1.16.1 Available under license

1.17 sirupsen/logrus 1.9.3

1.17.1 Available under license

1.18 go-multierror 1.1.1

1.18.1 Available under license

1.19 backoff 4.3.0

1.19.1 Available under license

1.20 mitchellh-mapstructure 1.5.0

1.20.1 Available under license

1.21 zstd 1.3.1

1.21.1 Available under license

1.22 go-retryablehttp 0.7.7

1.22.1 Available under license

1.23 readline 8.2.13-r0

1.23.1 Available under license

1.24 alpine-keys 2.5-r0

1.24.1 Available under license

1.25 go-rootcerts 1.0.2

1.25.1 Available under license

1.26 openssl 3.0.15

1.26.1 Available under license

1.27 pam 1.6.1-r1

1.27.1 Available under license

1.28 zlib 1.3.1-r2

1.28.1 Available under license

1.29 bash 5.2.37-r0

1.29.1 Available under license

1.30 pax-utils 1.3.8-r1

1.30.1 Available under license

1.31 go-sockaddr 1.0.7

1.31.1 Available under license

1.32 skalibs 2.14.3.0-r0

1.32.1 Available under license

1.33 utmps 0.1.2.3-r2

1.33.1 Available under license
1.34 ncurses 6.5_p20241006-r3
1.34.1 Available under license
1.35 libcap 2.71-r0
1.35.1 Available under license
1.36 iputils 20240905-r0
1.36.1 Available under license
1.37 shadow 4.16.0-r1
1.37.1 Available under license
1.38 libncursesw 6.5_p20241006-r3
1.38.1 Available under license
1.39 scanelf 1.3.8-r1
1.39.1 Available under license
1.40 utmps-libs 0.1.2.3-r2
1.40.1 Available under license
1.41 skalibs-libs 2.14.3.0-r0
1.41.1 Available under license
1.42 auto-sdk 1.1.0
1.42.1 Available under license
1.43 iputils-clockdiff 20240905-r0
1.43.1 Available under license
1.44 iputils-tracepath 20240905-r0
1.44.1 Available under license
1.45 iputils-ping 20240905-r0
1.45.1 Available under license
1.46 iputils-arping 20240905-r0
1.46.1 Available under license
1.47 alpine-baselayout 3.6.8-r1
1.47.1 Available under license
1.48 musl 1.2.5-r9
1.48.1 Available under license
1.49 stdr 1.2.2
1.49.1 Available under license
1.50 go-glob 1.0.0
1.50.1 Available under license
1.51 go-secure-stdlib-strutil 0.1.2
1.51.1 Available under license
1.52 ca-certificates 20241121-r1
1.52.1 Available under license

1.53 busybox 1.37.0-r12

1.53.1 Available under license

1.54 ssl-client 1.37.0-r12

1.54.1 Available under license

1.55 s2a-go 0.1.9

1.55.1 Available under license

1.56 ca-certificates-bundle 20241121-r1

1.56.1 Available under license

1.57 alpine-baselayout-data 3.6.8-r1

1.57.1 Available under license

1.58 github.com/google/s2a-go 0.1.9

1.58.1 Available under license

1.59 go-cleanhttp 0.5.2

1.59.1 Available under license

1.60 busybox 1.37.0-r14

1.60.1 Available under license

1.61 openssl 3.3.3-r0

1.61.1 Available under license

1.62 libcrypto3 3.3.3-r0

1.62.1 Available under license

1.63 libssl3 3.3.3-r0

1.63.1 Available under license

1.64 alpine-base 3.21.3-r0

1.64.1 Available under license

1.65 musl-utils 1.2.5-r9

1.65.1 Available under license

1.66 apk-tools 2.14.6-r3

1.66.1 Available under license

1.67 golang-genproto 0.0.0

1.67.1 Available under license

1.68 go.opentelemetry.io/auto/sdk 1.1.0

1.68.1 Available under license

1.69 github.com/go-logr/stdr 1.2.2

1.69.1 Available under license

1.70 alpine-release 3.21.3-r0

1.70.1 Available under license

1.71 vault-api 1.16.0

1.71.1 Available under license

1.72 x-time 0.11.0

- 1.72.1 Available under license
- 1.73 x-time-rate 0.11.0**
 - 1.73.1 Available under license
- 1.74 go-secure-stdlib-parseutil 0.2.0**
 - 1.74.1 Available under license
- 1.75 github.com/googleapis/enterprise-certificate-proxy 0.3.6**
 - 1.75.1 Available under license
- 1.76 enterprise-certificate-proxy 0.3.6**
 - 1.76.1 Available under license
- 1.77 github.com/hashicorp/go-secure-stdlib/parseutil 0.2.0**
 - 1.77.1 Available under license
- 1.78 protobuf-go 1.36.6**
 - 1.78.1 Available under license
- 1.79 protobuf 1.36.6**
 - 1.79.1 Available under license
- 1.80 logrus 1.9.3**
 - 1.80.1 Available under license
- 1.81 perks 1.0.1**
 - 1.81.1 Available under license
- 1.82 prometheus-client 1.22.0**
 - 1.82.1 Available under license
- 1.83 clientmodel 0.6.2**
 - 1.83.1 Available under license
- 1.84 prometheus-client-model 0.6.2**
 - 1.84.1 Available under license
- 1.85 github.com/go-jose/go-jose/v4 4.1.0**
 - 1.85.1 Available under license
- 1.86 cloud.google.com/go/auth/oauth2adapt 0.2.8**
 - 1.86.1 Available under license
- 1.87 go-jose-go-jose 4.1.0**
 - 1.87.1 Available under license
- 1.88 confluentinc-confluent-kafka-go 2.10.0**
 - 1.88.1 Available under license
- 1.89 procfs 0.16.1**
 - 1.89.1 Available under license
- 1.90 github.com/confluentinc/confluent-kafka-go/v2 2.10.0**
 - 1.90.1 Available under license
- 1.91 golang 1.24.3**
 - 1.91.1 Notifications

1.91.2 Available under license

1.92 x-net 0.40.0

1.92.1 Available under license

1.93 x-sys 0.33.0

1.93.1 Available under license

1.94 x-text 0.25.0

1.94.1 Available under license

1.95 stdlib go1.24.3

1.95.1 Available under license

1.96 x-crypto 0.38.0

1.96.1 Available under license

1.97 go-auth-oauth2adapt 0.2.8

1.97.1 Available under license

1.98 x-oauth2 0.30.0

1.98.1 Available under license

1.99 gax-go 2.14.2

1.99.1 Available under license

1.100 go-compute-metadata 0.7.0

1.100.1 Available under license

1.101 prometheus-common 0.64.0

1.101.1 Available under license

1.102 googlecloudgocomputemetadadata 0.7.0

1.102.1 Available under license

1.103 grpc-go 1.72.2

1.103.1 Available under license

1.104 google-golang-org-grpc 1.72.2

1.104.1 Available under license

1.105 common 0.64.0

1.105.1 Available under license

1.106 logr 1.4.3

1.106.1 Available under license

1.107 google-api-go-client 0.236.0

1.107.1 Available under license

1.108 otel 1.36.0

1.108.1 Available under license

1.109 cloud.google.com/go/auth 0.16.2

1.109.1 Available under license

1.110 google-golang-org-api 0.236.0

1.110.1 Available under license

1.111 google.golang.org/genproto/googleapis/rpc 0.0.0-20250603155806-513f23925822

1.111.1 Available under license

1.112 go.opentelemetry.io/otel/trace 1.36.0

1.112.1 Available under license

1.113 google-cloud-go-auth 0.16.2

1.113.1 Available under license

1.114 [otel-metric](#) 1.36.0

1.114.1 Available under license

1.115 [otel-trace](#) 1.36.0

1.115.1 Available under license

1.116 [bobesa-go-domain-util](#) 0.0.0-20250410211237-17ab3b2f4a95

1.116.1 Available under license

1.117 [genproto-googleapis-rpc](#) 0.0.0-20250603155806-513f23925822

1.117.1 Available under license

1.118 github.com/bobesa/go-domain-util 0.0.0-20250410211237-17ab3b2f4a95

1.118.1 Available under license

1.119 [contrib-instrumentation-net-http-otelhttp](#) 0.61.0

1.119.1 Available under license

1.120 [librdkafka](#) 2.10.0-r0

1.120.1 Notifications

1.120.2 Available under license

1.121 github.com/google/uuid 1.6.0

1.121.1 Available under license

1.122 [cjson](#) 1.7.18-r1

1.122.1 Notifications

1.122.2 Available under license

1.123 [curl](#) 8.14.1-r1

1.123.1 Available under license

1.1 lz4 1.3.1

1.1.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third

parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further

restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```


`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General Public License instead of this License.

Copyright (c) 2014, lpsantil

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This repository uses 2 different licenses :

- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPLv2 license, unless explicitly stated otherwise

Relevant license is reminded at the top of each source file, and with presence of COPYING or LICENSE file in associated directories.

This model is selected to emphasize that files in the `lib` directory are designed to be included into 3rd party applications, while all other files, in `programs`, `tests` or `examples`, receive more limited attention and support for such scenario.

LZ4 Library

Copyright (c) 2011-2016, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: liblz4

Upstream-Contact: Yann Collet <Cyan4973@github.com>

Source: <https://github.com/lz4/lz4>

Files: *

Copyright: (C) 2011+ Yann Collet

License: GPL-2+

The full text of license: <https://github.com/Cyan4973/lz4/blob/master/lib/LICENSE>

1.2 errwrap 1.1.0

1.2.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable

by such Contributor that

would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)

ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3.

Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent

Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of

either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained

within the Source Code Form of the Covered

Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an

ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections

5.1 or 5.2 above, all end user

license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

1.3 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.3.1 Available under license :

Copyright (c) 2011, Open Knowledge Foundation Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.4 kelseyhightower-envconfig 1.4.0

1.4.1 Available under license :

Copyright (c) 2013 Kelsey Hightower

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 hcl 1.0.0

1.5.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the

creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable

by such Contributor that

would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)

ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)

Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3.

Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent

Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions

are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered

Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained

within the Source Code Form of the Covered

Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing

basis,

if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections

5.1 or 5.2 above, all end user

license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation.

Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With

Secondary Licenses under the terms of this version of the License,
the
notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the
terms of the Mozilla Public License, v.
2.0. If a copy of the MPL was not
distributed with this file, You can
obtain one at
<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then
You may include the notice in a location (such as a LICENSE file in a relevant
directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible
With Secondary Licenses, as defined by
the Mozilla Public License, v. 2.0.

1.6 open-telemetry-opentelemetry-collector-contrib 0.61.0

1.6.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.7 githubcommitchellhmapstructure 1.5.0

1.7.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 Mitchell Hashimoto

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.8 httpsnoop 1.0.4

1.8.1 Available under license :

Copyright (c) 2016 Felix Geisendrfer (felix@debuggable.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.9 libmd 1.1.0-r0

1.9.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/sha2.h

src/sha2.c

Copyright:

Copyright 2000-2001, Aaron D. Gifford

All rights reserved.

License: BSD-3-clause-Aaron-D-Gifford

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/rmd160.h

src/rmd160.c

Copyright:

Copyright 2001 Markus Friedl. All rights reserved.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

src/md2.c

Copyright:

Copyright (c) 2001 The NetBSD Foundation, Inc.

All rights reserved.

.

This code is derived from software contributed to The NetBSD Foundation by Andrew Brown.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/rmd160.3

man/sha1.3

man/sha2.3

Copyright:

Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>

License: ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

man/mdX.3

src/helper.c

Copyright:

Poul-Henning Kamp <phk@login.dkuug.dk>

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this file. As long as you retain
this notice you

can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

Files:

include/md4.h

src/md4.c

Copyright:

Colin Plumb

Todd C. Miller

License: public-domain-md4

This code implements the MD4 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:

include/md5.h

src/md5.c

Copyright:

Colin Plumb

License: public-domain-md5

This code implements the MD5 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Files:

include/sha1.h

src/sha1.c

Copyright:

Steve Reid <steve@edmweb.com>

License: public-domain-sha1

100% Public

1.10 zlib 1.3.1

1.10.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.11 google-uuid 1.6.0

1.11.1 Available under license :

Paul Borman <borman@google.com>

bmatsuo

shawnps

theory

jboverfelt

dsymonds

cd1

wallclockbuilder

dansouza

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.12 libbsd 0.12.2-r0

1.12.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Files:

include/bsd/err.h

include/bsd/stdlib.h

include/bsd/sys/param.h

include/bsd/unistd.h

src/bsd_getopt.c

src/err.c

src/fgetln.c

src/progname.c

Copyright:

Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>

Copyright 2005 Hector Garcia Alvarez

Copyright 2005 Aurelien Jarno

Copyright 2006 Robert Millan

Copyright 2018 Facebook, Inc.

License: BSD-3-clause

Files:

include/bsd/netinet/ip_icmp.h

include/bsd/sys/bitstring.h

include/bsd/sys/queue.h

include/bsd/sys/time.h

include/bsd/timeconv.h

include/bsd/vis.h

man/bitstring.3bsd

man/errc.3bsd

man/explicit_bzero.3bsd

man/fgetln.3bsd

man/fgetwln.3bsd

man/fpurge.3bsd

man/funopen.3bsd

man/getbsize.3bsd

man/heapsort.3bsd

man/nlist.3bsd

man/pwcache.3bsd

man/queue.3bsd
man/radixsort.3bsd
man/reallocarray.3bsd
man/reallocf.3bsd
man/setmode.3bsd
man/strmode.3bsd
man/strnstr.3bsd
man/strtoi.3bsd
man/strtou.3bsd
man/unvis.3bsd
man/vis.3bsd
man/wcslcpy.3bsd

src/getbsize.c
src/heapsort.c
src/merge.c
src/nlist.c
src/pwcache.c
src/radixsort.c
src/setmode.c
src/strmode.c
src/strnstr.c
src/strtoi.c
src/strtou.c
src/unvis.c

Copyright:

Copyright 1980, 1982, 1986, 1989-1994

The Regents of the University of California. All rights reserved.

Copyright 1992 Keith Muller.

Copyright 2001 Mike Barcroft <mike@FreeBSD.org>

.

Some code is derived from software contributed to Berkeley by the American National Standards Committee X3, on Information Processing Systems.

.

Some code is derived from software contributed to Berkeley by Peter McIlroy.

.

Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

.

Some code is derived from software contributed to Berkeley by Dave Borman at Cray Research, Inc.

.

Some code is derived from software contributed to Berkeley by Paul Vixie.

.

Some code is derived from software contributed to Berkeley by

Chris Torek.

.

Copyright UNIX System Laboratories, Inc.

All or some portions of this file are derived from material licensed to the University of California by American Telephone and Telegraph Co. or Unix System Laboratories, Inc. and are reproduced herein with the permission of UNIX System Laboratories, Inc.

License: BSD-3-clause-Regents

Files:

src/vis.c

Copyright:

Copyright 1989, 1993

The Regents of the University of California. All rights reserved.

.

Copyright 1999, 2005 The NetBSD Foundation, Inc.

All rights reserved.

License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:

include/bsd/libutil.h

Copyright:

Copyright 1996 Peter Wemm <peter@FreeBSD.org>.

All rights reserved.

Copyright 2002 Networks Associates Technology, Inc.

All rights reserved.

License: BSD-3-clause-author

Files:

man/timeradd.3bsd

Copyright:

Copyright

2009 Jukka Ruohonen <jruohonen@iki.fi>

Copyright 1999 Kelly Yancey <kbyanc@posi.net>

All rights reserved.

License: BSD-3-clause-John-Birrell

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.

THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/setproctitle.3bsd

Copyright:

Copyright 1995 Peter Wemm <peter@FreeBSD.org>

All rights reserved.

License: BSD-5-clause-Peter-Wemm

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author Peter Wemm.
5. Modifications may be freely made to this file providing the above conditions are met.

Files:

include/bsd/stringlist.h

man/arc4random.3bsd

man/fmtcheck.3bsd

man/humanize_number.3bsd

man/stringlist.3bsd

man/timeval.3bsd

src/fmtcheck.c

src/humanize_number.c

src/stringlist.c

src/strtonum.c

Copyright:

Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014

The NetBSD Foundation, Inc.

Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>

All rights reserved.

.

Copyright 2014

The NetBSD Foundation, Inc.

All rights reserved.

.

Some code was derived from software contributed to The NetBSD Foundation by Taylor R. Campbell.

.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.

.

Some code is derived from software contributed to The NetBSD Foundation by Jason R. Thorpe of the Numerical Aerospace Simulation Facility, NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.

.

Some code is derived from software contributed to The NetBSD Foundation by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

.

Some code is derived from software contributed to The NetBSD Foundation by Jukka Ruohonen.

License: BSD-2-clause-NetBSD

Files:

include/bsd/sys/endian.h

man/byteorder.3bsd

man/closefrom.3bsd

man/expand_number.3bsd

man/flopen.3bsd

man/getpeereid.3bsd

man/pidfile.3bsd

src/expand_number.c

src/pidfile.c

src/reallocf.c

src/timeconv.c

Copyright:

Copyright 1998, M. Warner Losh <imp@freebsd.org>

All rights reserved.

.

Copyright 2001 Dima Dorfman.

All rights reserved.

.

Copyright 2001 FreeBSD Inc.

All rights reserved.

.

Copyright 2002 Thomas Moestl <tmml@FreeBSD.org>

All rights reserved.

.

Copyright 2002 Mike Barcroft <mike@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Colin Percival

All rights reserved.

.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>

Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.

.

Copyright 2009 Advanced Computing Technologies LLC

All rights reserved.

.

Copyright 2011 Guillem Jover <guillem@hadrons.org>

License: BSD-2-clause

Files:

src/flopen.c

Copyright:

Copyright 2007-2009

Dag-Erling Codan Smrgrav

All rights reserved.

License: BSD-2-clause-verbatim

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/bsd/sys/tree.h

man/fparsi.n.3bsd

man/tree.3bsd

src/fparsi.n.c

Copyright:

Copyright 1997 Christos Zoulas.

All rights reserved.

.

Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

License: BSD-2-clause-author

Files:

include/bsd/readpassphrase.h

man/readpassphrase.3bsd

man/strlcpy.3bsd

man/strtonum.3bsd

src/arc4random.c

src/arc4random_linux.h

src/arc4random_uniform.c

src/arc4random_unix.h

src/arc4random_win.h

src/closefrom.c

src/freezero.c

src/getentropy_aix.c

src/getentropy_bsd.c

src/getentropy_hpux.c

src/getentropy_hurd.c

src/getentropy_linux.c

src/getentropy_osx.c

src/getentropy_solaris.c

src/getentropy_win.c

src/readpassphrase.c

src/reallocarray.c

src/recallocarray.c

src/strlcat.c

src/strncpy.c

test/explicit_bzero.c

test/strtonum.c

Copyright:

Copyright 2004 Ted Unangst and Todd Miller

All rights reserved.

.

Copyright 1996 David Mazieres <dm@uun.org>

Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015

Todd C. Miller <Todd.Miller@courtesan.com>

Copyright 2004 Ted Unangst

Copyright 2004 Otto Moerbeek <otto@drijf.net>

Copyright 2008 Damien Miller <djm@openbsd.org>

Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>

Copyright 2013 Markus Friedl <markus@openbsd.org>

Copyright 2014 Bob Beck <beck@obtuse.com>

Copyright 2014 Brent Cook <bcook@openbsd.org>

Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>

Copyright 2014 Theo de Raadt <deraadt@openbsd.org>

Copyright 2014 Google Inc.

Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>

License:

ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/inet_net_pton.c

Copyright:

Copyright 1996 by Internet Software Consortium.

License: ISC-Original

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice

and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS

ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/setproctitle.c

Copyright:

Copyright 2010 William Ahern

Copyright 2012 Guillem Jover <guillem@hadrons.org>

License: Expat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

src/explicit_bzero.c

src/chacha_private.h

Copyright:

None

License: public-domain

Public domain.

Files:

man/mdX.3bsd

Copyright:

None

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this
file. As long as you retain this notice you
can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

License: BSD-3-clause-author

Redistribution and use in source and binary forms, with or without
modification, is permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions
in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.13 xxhash 2.3.0

1.13.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.14 githubcomhashicorpcl 1.0.0

1.14.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable

by such Contributor that

would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)

ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as

part of a Larger Work; and

- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3.

Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent

Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at

their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance.

Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections

5.1 or 5.2 above, all end user

license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can

obtain one at

<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

1.15 github.com/hashicorp/go-secure-stdlib/strutil 0.1.2

1.15.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims,

in any patent Licensable by such

Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor

first

distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License.

Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in

Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices

(including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts

of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible

With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.16 goglob 1.0.0

1.16.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Ryan Uber

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.17 sirupsen-logrus 1.9.3

1.17.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.18 go-multierror 1.1.1

1.18.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributors Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable

by such Contributor that

would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)

ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3.

Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent

Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including

copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections

5.1 or 5.2 above, all end user

license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without

reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not

distributed with this file, You can
obtain one at
<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then
You may include the notice in a location (such as a LICENSE file in a relevant
directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible
With Secondary Licenses, as defined by
the Mozilla Public License, v. 2.0.

1.19 backoff 4.3.0

1.19.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Cenk Alt

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.20 mitchellh-mapstructure 1.5.0

1.20.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 Mitchell Hashimoto

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.21 zstd 1.3.1

1.21.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 2015 Sergey Dryabzhinsky

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions are met:

*

* 1. Redistributions of source code must retain the above copyright notice,

* this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright notice,

* this list of conditions and the following disclaimer in the documentation

* and/or other materials provided with the distribution.

*

* 3. Neither the name of Steeve Morin nor the names of its contributors may be

* used to endorse or promote products derived from this software without

* specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):

* /zstd-1.3.1-tar-gz/zstd-1.3.1/src/python-zstd.h

No license file was found, but licenses were detected in source scan.

/* *****

bitstream

Part of FSE library

header file (to include)

Copyright (C) 2013-2017, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without
 modification, are permitted provided that the following conditions are
 met:

* Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
 copyright notice, this list of conditions and the following disclaimer
 in the documentation and/or other materials provided with the
 distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE IMPLIED
 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- Source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/bitstream.h

No license file was found, but licenses were detected in source scan.

/**

* Copyright (c) 2016 Tino Reichardt

* All rights reserved.

*

* This source code is licensed under the BSD-style license found in the

* LICENSE file in the root directory of this source tree. An additional grant

* of patent rights can be found in the PATENTS file in the same directory.

*

* You can contact the author at:

* - zstdmt source repository: <https://github.com/mcmilk/zstdmt>

*/

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/threading.h

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/threading.c

No license file was found, but licenses were detected in source scan.

/*

* ZSTD Library Python bindings

* Copyright (c) 2015, Sergey Dryabzhinsky

* All rights reserved.

*

* BSD License

*

* Redistribution and use in source and binary forms, with or without modification,

* are permitted provided that the following conditions are met:

*

* * Redistributions of source code must retain the above copyright notice, this

* list of conditions and the following disclaimer.

*

* * Redistributions in binary form must reproduce the above copyright notice, this

* list of conditions and the following disclaimer in the documentation and/or

* other materials provided with the distribution.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS

BE LIABLE FOR

* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/* Macros and other changes from python-lz4.c

* Copyright (c) 2012-2013, Steeve Morin

* All rights reserved. */

Found in path(s):

* /zstd-1.3.1-tar-gz/zstd-1.3.1/src/python-zstd.c

No license file was found, but licenses were detected in source scan.

/* *****

Huffman decoder, part of New Generation Entropy library

Copyright (C) 2013-2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE+HUF source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/decompress/huf_decompress.c

No license file was found, but licenses were detected in source scan.

/* *****

FSE : Finite State Entropy codec

Public Prototypes declaration

Copyright (C) 2013-2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- Source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/fse.h

No license file was found, but licenses were detected in source scan.

/* *****

Huffman coder, part of New Generation Entropy library

header file

Copyright (C) 2013-2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- Source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/huf.h

No license file was found, but licenses were detected in source scan.

Metadata-Version: 1.1

Name: zstd

Version: 1.3.1

Summary: ZSTD Bindings for Python

Home-page: <https://github.com/sergey-dryabzhinsky/python-zstd>

Author: Sergey Dryabzhinsky

Author-email: sergey.dryabzhinsky@gmail.com

License: BSD

Description: =====

python-zstd

=====

.. image:: <https://travis-ci.org/sergey-dryabzhinsky/python-zstd.svg?branch=master>
:target: <https://travis-ci.org/sergey-dryabzhinsky/python-zstd>

Simple python bindings to Yann Collet ZSTD compression library

****Zstd****, short for Zstandard, is a new lossless compression algorithm, which provides both good compression ratio and speed for your standard compression needs. "Standard" translates into everyday situations which neither look for highest possible ratio (which LZMA and ZPAQ cover) nor extreme speeds (which LZ4 covers).

It is provided as a BSD-license package, hosted on GitHub_.

.. _GitHub: <https://github.com/facebook/zstd>

WARNING!!!

If you setup 1.0.0.99.1 version - remove it manually to able to update.
PIP matching version strings not tuple of numbers.

Result generated by versions prior to 1.0.0.99.1 is not compatible with original Zstd by any means. It generates custom header and can be readed only by zstd python module.

As of 1.0.0.99.1 version it uses standard Zstd output, not modified.

To prevent data loss there is two functions now: ``compress\_old`` and ``decompress\_old``. They are works just like in old versions prior to 1.0.0.99.1.

As of 1.1.4 version module build without them by default.

LINKS

* More full-featured and compatible with Zstandard python bindings
by Gregory Szorc: <https://pypi.python.org/pypi/zstandard>

TODO

* Support dictionary training. Current status is ``not supported``.

Build from source

```
>>> git clone https://github.com/sergey-dryabzhinsky/python-zstd
>>> git submodule update --init
>>> apt-get install python-dev python3-dev python-setuptools python3-setuptools
>>> python setup.py build_ext clean
>>> python3 setup.py build_ext clean
```

Note: legacy format support disabled by default.

To build with legacy support - pass ``--legacy`` option to setup.py script:

```
>>> python setup.py build_ext --legacy clean
```

If you want to build with existing distribution of libzstd just add ``--external`` option.

But beware! Legacy formats support is unknown in this case.

```
>>> python setup.py
build_ext --external clean
```

If paths to header file ``zstd.h`` and libraries is uncommon - use common ``build`` params:

```
--libraries --include-dirs --library-dirs.
```

```
>>> python setup.py build_ext --external --include-dirs /opt/zstd/usr/include --libraries zstd --library-dirs
/opt/zstd/lib clean
```

Install from pypi

```
>>> # for Python 2.6+
>>> pip install zstd
>>> # or for Python 3.2+
>>> pip3 install zstd
```

Keywords: zstd,zstandard,compression

Platform: UNKNOWN

Classifier: License :: OSI Approved :: BSD License

Classifier: Intended Audience :: Developers

Classifier: Development Status :: 5 - Production/Stable

Classifier: Operating System :: POSIX

Classifier: Programming Language :: C

Classifier: Programming Language :: Python

Classifier: Programming Language :: Python :: 2.6

Classifier: Programming Language :: Python :: 2.7

Classifier:

Programming Language :: Python :: 3.2

Classifier: Programming Language :: Python :: 3.3

Classifier: Programming Language :: Python :: 3.4

Classifier: Programming Language :: Python :: 3.5

Classifier: Programming Language :: Python :: 3.6

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/PKG-INFO

* /zstd-1-3-1-tar-gz/zstd-1.3.1/src/zstd.egg-info/PKG-INFO

No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 2016-present, Yann Collet, Facebook, Inc.

* All rights reserved.

*

* This source code is licensed under both the BSD-style license (found in the

* LICENSE file in the root directory of this source tree) and the GPLv2 (found

* in the COPYING file in the root directory of this source tree).

*/

/* *****

mem.h

low-level memory access routines

Copyright (C) 2013-2015, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice,

this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v06.c

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v03.c

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v04.c

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v02.c

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v07.c

No license file was found, but licenses were detected in source scan.

'License :: OSI Approved :: BSD License',

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/setup.py

No license file was found, but licenses were detected in source scan.

/* *****

Huffman encoder, part of New Generation Entropy library

Copyright (C) 2013-2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE+HUF source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/huf_compress.c

No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 2016-present, Yann Collet, Facebook, Inc.

* All rights reserved.

*

* This source code is licensed under both the BSD-style license (found in the

* LICENSE file in the root directory of this source tree) and the GPLv2 (found

* in the COPYING file in the root directory of this source tree).

*/

/* *****

mem.h

low-level memory access routines

Copyright (C) 2013-2015, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice,
this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSEv05 source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v05.c

No license file was found, but licenses were detected in source scan.

/* *****

FSE : Finite State Entropy decoder

Copyright (C) 2013-2015, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/fse_decompress.c

No license file was found, but licenses were detected in source scan.


```

/*
 * xxHash - Fast Hash algorithm
 * Copyright (C) 2012-2016, Yann Collet
 *
 * BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 *
 * * Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following disclaimer
 * in the documentation and/or other materials provided with the
 * distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
 * THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * You can contact the author at :
 * - xxHash homepage: http://www.xxhash.com
 * - xxHash source repository : https://github.com/Cyan4973/xxHash
 */

```

Found in path(s):

```
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/xxhash.c
```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (c) 2016-present, Yann Collet, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under both the BSD-style license (found in the
 * LICENSE file in the root directory of this source tree) and the GPLv2 (found
 * in the COPYING file in the root directory of this source tree).
 */
/*

```

zstd - standard compression library
Copyright (C) 2014-2015, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- zstd source repository : <https://github.com/Cyan4973/zstd>
- zstd public forum : <https://groups.google.com/forum/#!forum/lz4c>

*/

Found

in path(s):

- * /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v01.c

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (c) 2016-present, Yann Collet, Facebook, Inc.

- * All rights reserved.

*

- * This source code is licensed under both the BSD-style license (found in the
- * LICENSE file in the root directory of this source tree) and the GPLv2 (found
- * in the COPYING file in the root directory of this source tree).

*/

Found in path(s):

- * /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/error_private.c

```

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/zstdmt_compress.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/zstd_internal.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/pool.c
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v06.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/decompress/zstd_decompress.c
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/zstd_common.c
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v04.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/zstd_errors.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v07.h
*
/zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_legacy.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/mem.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v02.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/zstdmt_compress.c
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v01.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/zstd_opt.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/zstd_compress.c
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/error_private.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/zstd.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v03.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/pool.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/legacy/zstd_v05.h
* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/compiler.h

```

No license file was found, but licenses were detected in source scan.

```
/*
```

Common functions of New Generation Entropy library
Copyright (C) 2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT

OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE+HUF source repository : <https://github.com/Cyan4973/FiniteStateEntropy>
- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/entropy_common.c

No license file was found, but licenses were detected in source scan.

/*

xxHash - Extremely Fast Hash algorithm

Header File

Copyright (C) 2012-2016, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- xxHash source repository : <https://github.com/Cyan4973/xxHash>

*/

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/common/xxhash.h

No license file was found, but licenses were detected in source scan.

/* *****

FSE : Finite State Entropy encoder

Copyright (C) 2013-2015, Yann Collet.

BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :

- FSE source repository : <https://github.com/Cyan4973/FiniteStateEntropy>

- Public forum : <https://groups.google.com/forum/#!forum/lz4c>

***** */

Found in path(s):

* /zstd-1-3-1-tar-gz/zstd-1.3.1/zstd/lib/compress/fse_compress.c

1.22 go-retryablehttp 0.7.7

1.22.1 Available under license :

Copyright (c) 2015 HashiCorp, Inc.

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a

separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the

outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary

to comply with
the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by

reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4.

Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License

with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing,

repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward.
Except as provided in Section

10.3, no one other than the license steward has the right to modify or

publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.23 readline 8.2.13-r0

1.23.1 Available under license :

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,

@c hence no sectioning command or @node.

@display

Copyright @copyright{ } 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{<http://fsf.org/>}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that

contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file

format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `La@TeX{}` input format, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever

to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that

edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in

the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements'' or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original

author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of

Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract

a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole

or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License,

to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the

earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to
"keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This

License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the

written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object

code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above

requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a

patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
```

```
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free

program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three

years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each

time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among

countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.24 alpine-keys 2.5-r0

1.24.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=alpine-keys
pkgver=2.5
pkgrel=0
pkgdesc="Public keys for Alpine Linux packages"
url="https://alpinelinux.org"
# we install arch specific keys to /etc so we cannot do arch=noarch
arch="all"
license="MIT"
replaces="alpine-base"
```

1.25 go-rootcerts 1.0.2

1.25.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the

License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a

separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by

contract or

otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks,

or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the

Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect,

You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible

With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.26 openssl 3.0.15

1.26.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it,

that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of

transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
- b) use the modified Package only within your corporation or organization.
- c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
- b) accompany the distribution with the machine-readable source of the Package with your modifications.
- c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output

from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.27 pam 1.6.1-r1

1.27.1 Available under license :

Unless otherwise *explicitly* stated the following text describes the licensed conditions under which the contents of this Linux-PAM release may be distributed:

Redistribution and use in source and binary forms of Linux-PAM, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License, in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,

BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Unless otherwise *explicitly* stated the following text describes the licensed conditions under which the contents of this libpamc release may be distributed:

Redistribution and use in source and binary forms of libpamc, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Library General Public License (LGPL), in which case the provisions of the GNU LGPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU LGPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.28 zlib 1.3.1-r2

1.28.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.29 bash 5.2.37-r0

1.29.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

=====

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed

this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.

This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction

and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it

is

released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the

product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the

only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the

additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept

this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.

If the

Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you

to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General

Public License instead of this License. But first, please read

<<https://www.gnu.org/licenses/why-not-lgpl.html>>.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free

software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion

of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of

Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or

distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each

time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to

be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,

INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We

have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that

the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The

Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy

a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already

includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a

copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this

License. Any attempt

otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this

License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,

@c hence no sectioning command or @node.

@display

Copyright @copyright{ } 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free

program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain @sc{ascii} without markup, Texinfo input format, LaTeX input format, @acronym{SGML} or @acronym{XML} using a publicly available @acronym{DTD}, and standard-conforming simple @acronym{HTML}, PostScript or @acronym{PDF} designed for human modification. Examples of transparent image formats include @acronym{PNG}, @acronym{XCF} and @acronym{JPG}. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, @acronym{SGML} or @acronym{XML} for which the @acronym{DTD} and/or processing tools are not generally available, and the machine-generated @acronym{HTML}, PostScript or @acronym{PDF} produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent

copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and

list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of

the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have

received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco,

California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts

being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of

packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works

for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a

copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent

the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you

must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and

propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or

that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's

public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to

do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>. Unless otherwise stated, all files in this directory are Copyright (C) 1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003, 2004,2005,2006,2007,2008,2009,2010,2011 Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

1.30 pax-utils 1.3.8-r1

1.30.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute

verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can

be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the

entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program
(or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program
in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other

circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.31 go-sockaddr 1.0.7

1.31.1 Available under license :

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributors Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but

not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by

this License.

1.10. Modifications

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable

by such Contributor that

would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)

ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3.

Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent

Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the

Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained

within the Source Code Form of the Covered

Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor

are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis,

if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections

5.1 or 5.2 above, all end user

license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability

shall not apply to liability for death or personal injury resulting from such partys negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a partys ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.
Copyright (c) Yasuhiro MATSUMOTO <mattn.jp@gmail.com>

MIT License (Expat)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014 Armon Dadgar

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients

of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [2013] [the CloudFoundry Authors]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright (c) 2016 HashiCorp, Inc.

Mozilla Public License Version 2.0

=====

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was

made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or

(b) any new file in Source Code Form that contains any Covered
Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including
without limitation, method,
process, and apparatus claims, in any patent Licensable by such
Contributor that would be infringed, but for the grant of the
License, by the making, using, selling, offering for sale, having
made, import, or transfer of either its Contributions or its
Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU
Lesser General Public License, Version 2.1, the GNU Affero General
Public License, Version 3.0, or any later versions of those
licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- (b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its

Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License.

You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- (a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- (b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5.

Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support,

indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```

*****
*
*
*
* 6. Disclaimer of Warranty
* -----
*
* Covered Software is provided under this License on an "as is"
* basis, without warranty of any kind, either expressed, implied, or
* statutory, including, without limitation, warranties that the
* Covered Software is free of defects, merchantable, fit for a
* particular purpose or non-infringing. The entire risk as to the
* quality and performance of the Covered Software is with You.
* Should any Covered Software prove defective in any respect, You
* (not any Contributor) assume the cost of any necessary servicing,
* repair, or correction. This disclaimer of warranty constitutes an
* essential part of
this License. No use of any Covered Software is
* authorized under this License except under this disclaimer.
*
*****

*****
*
*
* 7. Limitation of Liability
* -----
*
* Under no circumstances and under no legal theory, whether tort
* (including negligence), contract, or otherwise, shall any
* Contributor, or anyone who distributes Covered Software as
* permitted above, be liable to You for any direct, indirect,
* special, incidental, or consequential damages of any character
*
* including, without limitation, damages for lost profits, loss of
* goodwill, work stoppage, computer failure or malfunction, or any
* and all other commercial damages or losses, even if such party
* shall have been informed of the possibility of such damages. This
* limitation of liability shall not apply to liability for death or
* personal injury resulting from such party's negligence to the
* extent applicable law prohibits such limitation. Some
* jurisdictions do not allow the exclusion or limitation of
* incidental or consequential damages, so this exclusion and
* limitation may not apply to You.
*
*****

8. Litigation
-----

```

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With

Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Copyright (c) 2016 Ryan Uber

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Mitchell Hashimoto

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.32 skalibs 2.14.3.0-r0

1.32.1 Available under license :

Copyright (c) 2011-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.33 utmps 0.1.2.3-r2

1.33.1 Available under license :

Copyright (c) 2018-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.34 ncurses 6.5_p20241006-r3

1.34.1 Available under license :

Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: aclocal.m4 package

Copyright: 2003-2019,2020 by Thomas E. Dickey

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent 'make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source <https://invisible-island.net/ncurses/ncurses.html>
This package is used for testing builds of ncurses.

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: acllocal.m4 package

Copyright: 1996-2020,2021 by Thomas E. Dickey

Licence: X11

Files: doc/html/NCURSES-Programming-HOWTO.html

Copyright: 2001 by Pradeep Padala

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files:

install-sh

Copyright: 1994 X Consortium

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or

sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Files: progs/tset.c ncurses/tinfo/read_termcap.c
Copyright: 1980,1991,1992,1993 The Regents of the University of California.
License: BSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list

of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGE.

-- vile: txtmode file-encoding=utf-8

Copyright 2018-2020,2021 Thomas E. Dickey

Copyright 1998-2017,2018 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

-- vile:txtmode fc=72

-- \$Id: COPYING,v 1.10 2021/01/01 09:54:30 tom Exp \$

Upstream source <https://invisible-island.net/ncurses/Ada95.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: aclocal.m4 package

Copyright: 2010-2020,2021 by Thomas E. Dickey

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a

copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh

Copyright: 1994 X Consortium

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except

as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent 'make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: aclocal.m4 package

Copyright: 2003-2020,2021 by Thomas E. Dickey

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,

DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh

Copyright: 1994 X Consortium

Licence: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent 'make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General

Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

1.35 libcap 2.71-r0

1.35.1 Available under license :

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/cap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/cap, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND

ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you;
rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components

(compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to

refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information
on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

```
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */
```

Unless otherwise **explicitly** stated, the following text describes the
licensed conditions under which the contents of this libcap/psx release
may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/psx, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above

restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This
General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if
you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License

along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

```
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */
```

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using

it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free

use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any

part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition,
mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium

customarily
used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If
distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you
do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented

by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new

versions

of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public

License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

/* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later */

Unless otherwise **explicitly** stated the following text describes the licensed conditions under which the contents of this module release may be distributed:

Redistribution and use in source and binary forms of this module, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Library General Public License, in which case the provisions of the GNU LGPL are required **INSTEAD OF** the above restrictions. (This clause is necessary due to a potential conflict between the GNU LGPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS

OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.36 iputils 20240905-r0

1.36.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any

patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any

part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium

customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions

of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.37 shadow 4.16.0-r1

1.37.1 Available under license :

SPDX-License-Identifier: BSD-3-Clause

All files under this project either

1. fall under the BSD 3 clause license (by default).
2. carry an SPDX header declaring what license applies.

or

3. list a full custom license

This software is originally

* Copyright (c) 1989 - 1994, Julianne Frances Haugh

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

- * 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
- * 3. The name of the copyright holders or contributors may not be used to
* endorse or promote products derived from this software without
* specific
prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contributor: Patrycja Rosa <alpine@ptrcnull.me>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=bsd-compat-headers

pkgver=0.7.2

pkgrel=6

pkgdesc="BSD compatibility headers (cdefs, queue, tree)"

url="https://gitlab.alpinelinux.org/alpine/aports"

arch="noarch"

license="BSD-2-Clause AND BSD-3-Clause"

source="

cdefs.h

queue.h

tree.h

"

builddir="\$srcdir"

options="!check" # just headers

package() {

mkdir -p "\$pkgdir"

install -Dm644 -t "\$pkgdir"/usr/include/sys \

cdefs.h queue.h tree.h

}

sha512sums="

37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312

7006442c3f019b6447a620c0cae73178 cdefs.h

2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86

11ea33987d4faba37dde5ce6da0750 queue.h

d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da

cef2092881eded0a2e94f6a07752ef3 tree.h

"

Automatically generated by apkbuild-cpan, template 4

Contributor: Valery Kartel <valery.kartel@gmail.com>

Maintainer: Celeste <cielesti@protonmail.com>

maintainer="Celeste <cielesti@protonmail.com>"

pkgname=perl-bsd-resource

pkgver=1.2911

pkgrel=10

```

#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/${_pkgreal}-${pkgver}"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gp517 py3-wheel"

```

```

subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

# License

All new source code committed to the shadow project is assumed to be made
available under the [BSD-3-Clause](../COPYING) license unless the submitter
specifies another license at that time. The shadow maintainers reserve the
right to refuse a submission if the license is deemed incompatible with the
goals of the project.

**Note**: old code may be made available under another license, check the
license tag for each file to get additional information.
# Contributor: Dhruvin Gandhi <contact@dhruvin>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.0
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"

```

```

builddir="$srcdir/${_pkgname}-${pkgver}"

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}

package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}

sha512sums="
cc7cd6b5bd8ea4c99b66f074871ffd7cc78b3e8b1b0d3afd9b95efdb7147c8293db2fac977d684fc34d3d60bd79066a06
dc2bff0a38b130d9d83c63538502db1
    license_expression-30.4.0.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-${pkgver}.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        --sysconfdir=/etc \

```

```

--mandir=/usr/share/man \
--scoredir=/usr/share/$pkgname
make
}

package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.24.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in $_types; do
subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
mkdir -p "$pkgdir"
}

_subpkg() {
local type=${subpkgname/$pkgname-/}
pkgdesc="$pkgdesc ($type)"
install_if="$pkgname=$pkgver-r$pkgrel"
mkdir -p "$subpkgdir"/usr/share/spdx
cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

```



```
list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt;
    do
        local license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}

sha512sums="
6eb81424e1b3173dbd8e17859b2e15dfe1c2af8f4da59b7dad0729c038a4cbdba95c34c2c546285fcdbbd41562e39575e
6a6ebb9463b70121e0a12b99d1307e7  license-list-data-3.24.0.tar.gz
"

mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.38 libncursesw 6.5_p20241006-r3

1.38.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
# ncurses 6.5 - patch 20241006 - Thomas E. Dickey
#
# -----
#
# Ncurses 6.5 is at
# https://invisible-island.net/archives/ncurses/
# https://invisible-mirror.net/archives/ncurses/
# https://ftp.gnu.org/gnu/ncurses/
#
# Patches for ncurses 6.5 can be found at
# https://invisible-island.net/archives/ncurses/6.5
# https://invisible-mirror.net/archives/ncurses/6.5
#
# -----
# https://invisible-island.net/archives/ncurses/6.5/ncurses-6.5-20241006.patch.gz
# patch by Thomas E. Dickey <dickey@invisible-island.net>
# created Sun Oct 6 23:44:01 UTC 2024
# -----
# NEWS | 6 +++
# VERSION | 2 -
# c++/cursesf.cc | 6 +--
# c++/cursesf.h
# | 6 +--
# c++/cursesm.cc | 6 +--
# c++/cursesm.h | 6 +--
# c++/cursespc.cc | 6 +--
# c++/cursespad.cc | 6 +--
# c++/cursslk.cc | 12 +++----
# c++/cursslk.h | 6 +--
# dist.mk | 4 +-
# misc/terminfo.src | 23 ++++++++---
# ncurses/base/lib_screen.c | 4 +-
# ncurses/base/new_pair.c | 4 +-
# ncurses/curses.priv.h | 4 +-
# ncurses/new_pair.h | 4 +-
# ncurses/tinfo/trim_sgr0.c | 7 ++--
# ncurses/tty/lib_twait.c | 6 +--
# package/debian-mingw/changelog | 4 +-
# package/debian-mingw64/changelog | 4 +-
# package/debian/changelog | 4 +-
# package/mingw-ncurses.nsi | 4 +-
# package/mingw-ncurses.spec | 2 -
# package/ncurses.spec | 2 -
# package/ncurset.spec |
```

2 -

# test/blue.c	10 +++---
# test/bs.c	8 ++--
# test/cardfile.c	12 +++----
# test/chgat.c	6 +--
# test/clip_printw.c	8 ++--
# test/color_name.h	6 +--
# test/demo_defkey.c	4 +-
# test/demo_forms.c	22 ++++++-----
# test/demo_menus.c	32 ++++++++-----
# test/demo_new_pair.c	4 +-
# test/demo_panels.c	16 ++++-----
# test/demo_termcap.c	18 +++++-----
# test/demo_terminfo.c	18 +++++-----
# test/dots.c	6 +--
# test/dots_mvcur.c	6 +--
# test/dots_termcap.c	8 ++--
# test/dup_field.c	14 ++++-----
# test/filter.c	6 +--
# test/form_driver_w.c	6 +--
# test/gdc.c	10 +++---
# test/inch_wide.c	6 +--
# test/inchs.c	6 +--
# test/ins_wide.c	6 +--
# test/insdelln.c	8 ++--
# test/inserts.c	6 +--
# test/knight.c	7 ++--
# test/list_keys.c	8 ++--
# test/move_field.c	18 +++++-----
# test/movewindow.c	16 ++++-----
# test/ncurses.c	61 ++++++++-----
# test/padview.c	12 +++-----
# test/parse_rgb.h	6 +--
# test/picsmap.c	14 ++++-----
# test/popup_msg.c	8 ++--
# test/popup_msg.h	6 +--
# test/railroad.c	6 +--
# test/rain.c	20 ++++++-----
# test/redraw.c	6 +--
# test/savescreen.c	14 ++++-----
# test/tclock.c	4 +-
# test/test.priv.h	10 +++---
# test/test_add_wchstr.c	6 +--
# test/test_addchstr.c	6 +--
# test/test_addstr.c	6 +--
# test/test_addwstr.c	6 +--

```
# test/test_endwin.c      | 6 +--
# test/test_get_wstr.c    | 8 ++--
# test/test_getstr.c      | 8 ++--
# test/test_instr.c       | 8 ++--
# test/test_inwstr.c      | 8 ++--
# test/test_mouse.c       | 4 +-
# test/test_opaque.c      | 8 ++--
# test/test_sgr.c         | 26 ++++++-----
# test/test_tparm.c       | 12 +++----
# test/test_vidputs.c     | 6 +--
# test/testcurs.c         | 8 ++--
# test/testscanw.c        | 6 +--
# test/view.c             | 21 ++++++-----
# 83 files changed, 392 insertions(+), 369 deletions(-)
```

```
# -----
```

Index: NEWS

Prereq:

```
1.4185
--- ncurses-6.5-20240928+/NEWS 2024-09-28 20:27:44.000000000 +0000
+++ ncurses-6.5-20241006/NEWS 2024-10-06 08:53:45.000000000 +0000
@@ -26,7 +26,7 @@
-- sale, use or other dealings in this Software without prior written --
-- authorization. --
```

```
-----
```

```
--- $Id: NEWS,v 1.4185 2024/09/28 20:27:44 tom Exp $
+-- $Id: NEWS,v 1.4188 2024/10/06 08:53:45 tom Exp $
```

```
-----
```

This is a log of changes that ncurses has gone through since Zeyd started

```
@@ -46,6 +46,10 @@
```

Changes through 1.9.9e did not credit all contributions;

it is not possible to add this information.

+20241006

+ + fixes for compiler warnings/cppcheck.

+ + use xterm+alt+title in wezterm -TD

+

20240928

+ improve error-message from infocmp when a terminal entry cannot be
opened (patch by Branden Robinson).

Index:

VERSION

```
--- ncurses-6.5-20240928+/VERSION 2024-09-28 09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/VERSION 2024-10-06 08:53:45.000000000 +0000
@@ -1 +1 @@
```

```
-5:0:10 6.5 20240928
```

```
+5:0:10 6.5 20241006
```

Index: c++/cursesf.cc

```

Prereq: 1.26
--- ncurses-6.5-20240928+/c++/cursesf.cc 2021-04-17 18:11:08.000000000 +0000
+++ ncurses-6.5-20241006/c++/cursesf.cc 2024-10-05 22:47:12.000000000 +0000
@@ -1,6 +1,6 @@
// * this is for making emacs happy: -*-Mode: C++;-*-
/*****
- * Copyright 2019-2020,2021 Thomas E. Dickey                *
+ * Copyright 2019-2021,2024 Thomas E. Dickey                *
* Copyright 1998-2005,2011 Free Software Foundation, Inc.    *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -36,7 +36,7 @@
#include "cursesf.h"
#include "cursesapp.h"

-MODULE_ID("$Id: cursesf.cc,v 1.26 2021/04/17 18:11:08 tom Exp $")
+MODULE_ID("$Id: cursesf.cc,v 1.27 2024/10/05 22:47:12 tom Exp $")

NCursesFormField::~NCursesFormField () THROWS(NCursesException)
{
@@ -76,7 +76,7 @@

void NCursesForm::setDefaultAttributes()
{
- NCursesApplication* S = NCursesApplication::getApplication();
+ const NCursesApplication* S = NCursesApplication::getApplication();

    int n = count();
    if (n > 0) {
Index: c++/cursesf.h
Prereq: 1.39
--- ncurses-6.5-20240928+/c++/cursesf.h 2022-08-20 20:52:15.000000000 +0000
+++ ncurses-6.5-20241006/c++/cursesf.h 2024-10-05 22:47:12.000000000 +0000
@@ -1,7 +1,7 @@
// * This makes emacs happy -*-Mode: C++;-*-
// vile:cppmode
/*****
- * Copyright 2019-2021,2022 Thomas E. Dickey                *
+ * Copyright 2019-2022,2024 Thomas E. Dickey                *
* Copyright 1998-2012,2014 Free Software
Foundation, Inc.      *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -33,7 +33,7 @@
* Author: Juergen Pfeifer, 1997                                *
*****/

-// $Id: cursesf.h,v 1.39 2022/08/20 20:52:15 tom Exp $

```

```

+// $Id: cursesf.h,v 1.40 2024/10/05 22:47:12 tom Exp $

#ifndef NCURSES_CURSESF_H_incl
#define NCURSES_CURSESF_H_incl 1
@@ -384,7 +384,7 @@
}

inline void *get_user() {
- UserHook* uptr = reinterpret_cast<UserHook*> (::form_userptr (form));
+ const UserHook* uptr = reinterpret_cast<UserHook*> (::form_userptr (form));
    assert (uptr != 0 && uptr->m_back==this && uptr->m_owner==form);
    return uptr->m_user;
}
Index: c++/cursesm.cc
Prereq: 1.27
--- ncurses-6.5-20240928+/c++/cursesm.cc 2021-04-17 18:11:08.000000000 +0000
+++ ncurses-6.5-20241006/c++/cursesm.cc 2024-10-05
22:47:12.000000000 +0000
@@ -1,6 +1,6 @@
// * this is for making emacs happy: -*-Mode: C++;-*-
/*****
- * Copyright 2019-2020,2021 Thomas E. Dickey                *
+ * Copyright 2019-2021,2024 Thomas E. Dickey                *
* Copyright 1998-2011,2017 Free Software Foundation, Inc.    *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -36,7 +36,7 @@
#include "cursesm.h"
#include "cursesapp.h"

-MODULE_ID("$Id: cursesm.cc,v 1.27 2021/04/17 18:11:08 tom Exp $")
+MODULE_ID("$Id: cursesm.cc,v 1.28 2024/10/05 22:47:12 tom Exp $")

NCursesMenuItem::~NCursesMenuItem() THROWS(NCursesException)
{
@@ -168,7 +168,7 @@
void
NCursesMenu::setDefaultAttributes()
{
- NCursesApplication* S = NCursesApplication::getApplication();
+
const NCursesApplication* S = NCursesApplication::getApplication();
if (S) {
    ::set_menu_fore(menu, S->foregrounds());
    ::set_menu_back(menu, S->backgrounds());
}
Index: c++/cursesm.h
Prereq: 1.35
--- ncurses-6.5-20240928+/c++/cursesm.h 2022-08-20 20:52:15.000000000 +0000

```

```

+++ ncurses-6.5-20241006/c++/cursesm.h 2024-10-05 22:47:12.000000000 +0000
@@ -1,6 +1,6 @@
// * This makes emacs happy -*-Mode: C++;-*-
/*****

- * Copyright 2019-2020,2022 Thomas E. Dickey                *
+ * Copyright 2019-2022,2024 Thomas E. Dickey                *
* Copyright 1998-2012,2014 Free Software Foundation, Inc.    *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -32,7 +32,7 @@
*   Author: Juergen Pfeifer, 1997                               *
*****/

-// $Id: cursesm.h,v 1.35 2022/08/20 20:52:15 tom Exp $
+// $Id: cursesm.h,v 1.36 2024/10/05 22:47:12 tom Exp $

#ifndef NCURSES_CURSESM_H_incl
#define NCURSES_CURSESM_H_incl 1
@@ -245,7 +245,7 @@
}

inline void *get_user() {
-   UserHook* uptr = STATIC_CAST(UserHook*)(::menu_userptr (menu));
+   const UserHook* uptr = STATIC_CAST(UserHook*)(::menu_userptr (menu));
   assert (uptr != 0 && uptr->m_back==this && uptr->m_owner==menu);
   return uptr->m_user;
}
Index: c++/cursesp.cc
Prereq: 1.27
--- ncurses-6.5-20240928+/c++/cursesp.cc 2020-02-02 23:34:34.000000000 +0000
+++ ncurses-6.5-20241006/c++/cursesp.cc 2024-10-05 22:47:12.000000000 +0000
@@ -1,6 +1,6 @@
// * this is for making emacs happy: -*-Mode: C++;-*-
/*****

- * Copyright 2019,2020 Thomas E. Dickey                *
+ * Copyright 2019-2020,2024
Thomas E. Dickey                *
* Copyright 1998-2003,2005 Free Software Foundation, Inc.    *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -35,7 +35,7 @@
#include "internal.h"
#include "cursesp.h"

-MODULE_ID("$Id: cursesp.cc,v 1.27 2020/02/02 23:34:34 tom Exp $")
+MODULE_ID("$Id: cursesp.cc,v 1.28 2024/10/05 22:47:12 tom Exp $")

NCursesPanel* NCursesPanel::dummy = static_cast<NCursesPanel*>(0);

```

```

@@@ -64,7 +64,7 @@@
void
NCursesPanel::redraw()
{
- PANEL *pan;
+ const PANEL *pan;

    pan = ::panel_above(NULL);
    while (pan) {
Index: c++/cursespad.cc
Prereq: 1.18
--- ncurses-6.5-20240928+/c++/cursespad.cc 2020-02-02 23:34:34.000000000 +0000
+++ ncurses-6.5-20241006+/c++/cursespad.cc 2024-10-05 22:47:12.000000000 +0000
@@@ -1,6 +1,6 @@@
// * this is for making emacs happy: -*-Mode: C++;-*-
/*****
-
* Copyright 2020 Thomas E. Dickey
+ * Copyright 2020,2024 Thomas E. Dickey
* Copyright 1999-2012,2013 Free Software Foundation, Inc.
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@@ -36,7 +36,7 @@@

#include <cursesw.h>

-MODULE_ID("$Id: cursespad.cc,v 1.18 2020/02/02 23:34:34 tom Exp $")
+MODULE_ID("$Id: cursespad.cc,v 1.19 2024/10/05 22:47:12 tom Exp $")

NCursesPad::NCursesPad(int nlines, int ncols)
: NCursesWindow(),
@@@ -221,7 +221,7 @@@
void NCursesFramedPad::OnOperation(int pad_req)
{
    (void) pad_req;
- NCursesWindow* W = Win();
+ const NCursesWindow* W = Win();
    NCursesWindow* W2 = getWindow();

    if ((static_cast<NCursesWindow*>(0) != W) && (static_cast<NCursesWindow*>(0) != W2)) {
Index: c++/cursslk.cc
Prereq:
1.22
--- ncurses-6.5-20240928+/c++/cursslk.cc 2024-06-15 21:07:30.000000000 +0000
+++ ncurses-6.5-20241006+/c++/cursslk.cc 2024-10-05 19:36:22.000000000 +0000
@@@ -35,10 +35,10 @@@
#include "internal.h"

```



```

#include "cursesapp.h"

-MODULE_ID("$Id: cursslk.cc,v 1.22 2024/06/15 21:07:30 tom Exp $")
+MODULE_ID("$Id: cursslk.cc,v 1.23 2024/10/05 19:36:22 tom Exp $")

Soft_Label_Key_Set::Soft_Label_Key&
- Soft_Label_Key_Set::Soft_Label_Key::operator=(char *text)
+ Soft_Label_Key_Set::Soft_Label_Key::operator=(const char *text)
{
    delete[] label;
    size_t need = 1 + ::strlen(text);
@@ -111,11 +111,11 @@

void Soft_Label_Key_Set::activate_label(int i, bool bf) {
    if (!b_attrInit) {
-    NCursesApplication* A = NCursesApplication::getApplication();
+    const NCursesApplication* A = NCursesApplication::getApplication();
        if (A) attrset(A->labels());
        b_attrInit = TRUE;
    }
-    Soft_Label_Key& K = (*this)[i];
+    const Soft_Label_Key& K = (*this)[i];
    if (ERR==::slk_set(K.num,bf?K.label:"",K.format))
        Error("slk_set");
    noutrefresh();
@@ -124,12 +124,12 @@
void Soft_Label_Key_Set::activate_labels(bool bf)
{
    if (!b_attrInit) {
-    NCursesApplication* A = NCursesApplication::getApplication();
+    const NCursesApplication* A = NCursesApplication::getApplication();
        if (A) attrset(A->labels());
        b_attrInit = TRUE;
    }
    for(int i=1; i <= num_labels; i++) {
-    Soft_Label_Key& K = (*this)[i];
+    const Soft_Label_Key& K = (*this)[i];
        if (ERR==::slk_set(K.num,bf?K.label:"",K.format))
            Error("slk_set");
    }
Index: c++/cursslk.h
Prereq: 1.19
--- ncurses-6.5-20240928+/c++/cursslk.h 2021-04-17 18:11:08.000000000 +0000
+++ ncurses-6.5-20241006+/c++/cursslk.h 2024-10-05 22:47:12.000000000 +0000
@@ -1,7 +1,7 @@
// * this is for making emacs happy: -*-Mode: C++;-*-
// vile:cppmode
/*****

```

```

- * Copyright 2019-2020,2021 Thomas E. Dickey
    *
+ * Copyright 2019-2021,2024 Thomas E. Dickey
* Copyright 1998-2003,2005 Free Software Foundation, Inc.
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -33,7 +33,7 @@
* Author: Juergen Pfeifer, 1997
*****/

-// $Id: cursslk.h,v 1.19 2021/04/17 18:11:08 tom Exp $
+// $Id: cursslk.h,v 1.20 2024/10/05 22:47:12 tom Exp $

#ifndef NCURSES_CURSSLK_H_incl
#define NCURSES_CURSSLK_H_incl
@@ -62,7 +62,7 @@

public:
    // Set the text of the Label
-    Soft_Label_Key& operator=(char *text);
+    Soft_Label_Key& operator=(const char *text);

    // Set the Justification of the Label
    Soft_Label_Key& operator=(Justification just) {
Index: dist.mk
Prereq: 1.1634
--- ncurses-6.5-20240928+/dist.mk 2024-09-28
09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/dist.mk 2024-10-06 08:53:45.000000000 +0000
@@ -26,7 +26,7 @@
# use or other dealings in this Software without prior written
# authorization.
#####
-# $Id: dist.mk,v 1.1634 2024/09/28 09:53:21 tom Exp $
+# $Id: dist.mk,v 1.1636 2024/10/06 08:53:45 tom Exp $
# Makefile for creating ncurses distributions.
#
# This only needs to be used directly as a makefile by developers, but
@@ -38,7 +38,7 @@
# These define the major/minor/patch versions of ncurses.
NCURSES_MAJOR = 6
NCURSES_MINOR = 5
-NCURSES_PATCH = 20240928
+NCURSES_PATCH = 20241006

# We don't append the patch to the version, since this only applies to releases
VERSION = $(NCURSES_MAJOR).$(NCURSES_MINOR)
Index: misc/terminfo.src

```

```

--- ncurses-6.5-20240928+/misc/terminfo.src 2024-09-07 16:23:31.000000000
+0000
+++ ncurses-6.5-20241006/misc/terminfo.src 2024-10-05 16:04:17.000000000 +0000
@@ -6,8 +6,8 @@
# Report bugs and new terminal descriptions to
# bug-ncurses@gnu.org
#
-# $Revision: 1.1150 $
-# $Date: 2024/09/07 16:23:31 $
+# $Revision: 1.1153 $
+# $Date: 2024/10/05 16:04:17 $
#
# The original header is preserved below for reference. It is noted that there
# is a "newer" version which differs in some cosmetic details (but actually
@@ -8060,6 +8060,17 @@
# "wezterm is a terminal emulator with support for modern features
# such as fonts with ligatures, hyperlinks, tabs and multiple windows."
#
+# The documentation bears mention. It refers to the "ANSI" standard and
+# https://wez.furlong.org/wezterm/what-is-a-terminal.html#ansi-and-ecma-48
+# X3.64 (withdrawn long ago). A related website
+# https://github.com/wez/ecma48
+# states that ECMA-48 was issued in 1979 and not revised since. Actually that
+# was the second revision; the fifth revision in 1991 is current. The source
+#
code refers to the second revision in a half-dozen places. Further, there
+# are three times as many references to Wikipedia as to xterm's documentation.
+# The git commit comments in several instances hint at an incomplete reading
+# of the relevant standards.
+#
# wezterm-20240203-110809-5046fc22 tested with MacOS
# General:
# + initial screensize is now 80x24
@@ -8094,7 +8105,7 @@
# + poor (doesn't copy anyone, most of the results are wrong)
# vttest:
# + DA is VT5xx with sixel, selective erase, user windows, color
-# + DA2 is VT220 version 277
+# + DA2 is VT220 version 277, perhaps a reference to xterm #277
# + only the VT100 character set works, contrary to DA/DA2.
# + no NRCS, either
# + double-sized character work, with some debris
@@ -8136,7 +8147,7 @@
use=ansi+rep, use=ansi+sgrdim, use=bracketed+paste,
use=ecma+index, use=ecma+italics, use=ecma+strikeout,
use=report+version, use=vt220+cvis,
- use=xterm+256color2, use=xterm+alt1049,
+ use=xterm+256color2,

```

```

use=xterm+alt+title,
use=xterm+focus, use=xterm+pcc2, use=xterm+pce2,
use=xterm+pcf2, use=xterm+sl-alt, use=xterm+sm+1006,
use=xterm+tmux,
@@ -27644,4 +27655,8 @@
# 2024-09-07
# + update comments -TD
#
+# 2024-10-05
+# + use xterm+alt+title in wezterm -TD
+# + update comments -TD
+#
##### SHANTIH! SHANTIH! SHANTIH!
Index: ncurses/base/lib_screen.c
Prereq: 1.108
--- ncurses-6.5-20240928+/ncurses/base/lib_screen.c 2024-09-22 20:20:36.000000000 +0000
+++ ncurses-6.5-20241006/ncurses/base/lib_screen.c 2024-10-05 20:47:44.000000000 +0000
@@ -42,7 +42,7 @@
#define CUR SP_TERMTYPE
#endif

-MODULE_ID("$Id: lib_screen.c,v 1.108 2024/09/22 20:20:36 tom Exp $")
+MODULE_ID("$Id: lib_screen.c,v 1.109 2024/10/05 20:47:44 tom Exp $")

#define MAX_SIZE 0x3fff /* 16k is big enough for a window or pad */

@@ -943,7 +943,7 @@
* Replace a window covering the whole screen, i.e., newscr or curscr.
*/
static WINDOW *
-replace_window(WINDOW *target, FILE *source)
+replace_window(const
WINDOW *target, FILE *source)
{
WINDOW *result = getwin(source);
#if NCURSES_EXT_FUNCS
Index: ncurses/base/new_pair.c
Prereq: 1.24
--- ncurses-6.5-20240928+/ncurses/base/new_pair.c 2024-07-27 19:22:23.000000000 +0000
+++ ncurses-6.5-20241006/ncurses/base/new_pair.c 2024-10-05 20:51:29.000000000 +0000
@@ -61,7 +61,7 @@

#endif

-MODULE_ID("$Id: new_pair.c,v 1.24 2024/07/27 19:22:23 tom Exp $")
+MODULE_ID("$Id: new_pair.c,v 1.25 2024/10/05 20:51:29 tom Exp $")

#if NCURSES_EXT_COLORS

```

```

@@ -193,7 +193,7 @@
* pair table.
*/
NCURSES_EXPORT(void)
-_nc_reset_color_pair(SCREEN *sp, int pair, colorpair_t * next)
+_nc_reset_color_pair(SCREEN *sp, int pair, const colorpair_t * next)
{
    colorpair_t *last;

Index: ncurses/curses.priv.h
Prereq: 1.691
--- ncurses-6.5-20240928+/ncurses/curses.priv.h 2024-09-28 15:35:01.000000000 +0000
+++ ncurses-6.5-20241006/ncurses/curses.priv.h 2024-10-05 21:01:50.000000000 +0000
@@ -35,7 +35,7 @@
*****/

/*
- * $Id: curses.priv.h,v 1.691 2024/09/28 15:35:01 tom Exp $
+ * $Id: curses.priv.h,v 1.692 2024/10/05 21:01:50 tom Exp $
*
* curses.priv.h
*

@@ -2170,7 +2170,7 @@
extern NCURSES_EXPORT(int) _nc_read_termcap_entry (const char *const, TERMTYPE2 *const);
extern NCURSES_EXPORT(int) _nc_setup_tinfo(const char *, TERMTYPE2 *);
extern NCURSES_EXPORT(int) _nc_setupscreen (int, int, FILE *, int, int);
-extern NCURSES_EXPORT(int) _nc_timed_wait (SCREEN *, int, int, int * EVENTLIST_2nd(_nc_eventlist *));
+extern NCURSES_EXPORT(int) _nc_timed_wait (const SCREEN *, int, int, int *
EVENTLIST_2nd(_nc_eventlist *));
extern NCURSES_EXPORT(int) _nc_trans_string (char *, const char *);
extern NCURSES_EXPORT(void) _nc_init_termtype (TERMTYPE2 *const);
extern NCURSES_EXPORT(void) _nc_do_color (int, int, int, NCURSES_OUTC);
Index: ncurses/new_pair.h
Prereq: 1.14
--- ncurses-6.5-20240928+/ncurses/new_pair.h 2024-05-25 23:10:33.000000000 +0000
+++
ncurses-6.5-20241006/ncurses/new_pair.h 2024-10-05 20:57:47.000000000 +0000
@@ -34,7 +34,7 @@
/*
* Common type definitions and macros for new_pair.c, lib_color.c
*
- * $Id: new_pair.h,v 1.14 2024/05/25 23:10:33 tom Exp $
+ * $Id: new_pair.h,v 1.15 2024/10/05 20:57:47 tom Exp $
*/

#ifdef NEW_PAIR_H
@@ -116,7 +116,7 @@

```

```

#if NCURSES_EXT_FUNCS && NCURSES_EXT_COLORS
extern NCURSES_EXPORT(void)  _nc_copy_pairs(SCREEN*, colorpair_t*, colorpair_t*, int);
extern NCURSES_EXPORT(void)  _nc_free_ordered_pairs(SCREEN*);
-extern NCURSES_EXPORT(void)  _nc_reset_color_pair(SCREEN*, int, colorpair_t*);
+extern NCURSES_EXPORT(void)  _nc_reset_color_pair(SCREEN*, int, const colorpair_t*);
extern NCURSES_EXPORT(void)  _nc_set_color_pair(SCREEN*, int, int);
#else
#define _nc_free_ordered_pairs(sp) /* nothing */
Index: ncurses/tinfo/trim_sgr0.c
Prereq: 1.23
--- ncurses-6.5-20240928+/ncurses/tinfo/trim_sgr0.c 2024-07-27 19:22:23.000000000 +0000
+++ ncurses-6.5-20241006/ncurses/tinfo/trim_sgr0.c 2024-10-05
20:53:48.000000000 +0000
@@ -37,7 +37,7 @@

#include <tic.h>

-MODULE_ID("$Id: trim_sgr0.c,v 1.23 2024/07/27 19:22:23 tom Exp $")
+MODULE_ID("$Id: trim_sgr0.c,v 1.24 2024/10/05 20:53:48 tom Exp $")

#undef CUR
#define CUR tp->
@@ -237,12 +237,10 @@

    if (PRESENT(exit_attribute_mode)
        && PRESENT(set_attributes)) {
- bool found = FALSE;
char *on = set_attribute_9(tp, 1);
char *off = set_attribute_9(tp, 0);
char *end = strdup(exit_attribute_mode);
char *tmp;
- size_t i, j, k;

TR(TRACE_DATABASE, ("checking if we can trim sgr0 based on sgr"));
TR(TRACE_DATABASE, ("sgr0    %s", _nc_visbuf(end)));
@@ -255,6 +253,9 @@
    FreeIfNeeded(off);
} else if (similar_sgr(off, end)
    && !similar_sgr(off, on)) {
+ bool found = FALSE;
+ size_t i, j, k;
+
TR(TRACE_DATABASE, ("adjusting sgr(9:off) : %s", _nc_visbuf(off)));
result = off;
/*
Index: ncurses/tty/lib_twait.c
Prereq: 1.81
--- ncurses-6.5-20240928+/ncurses/tty/lib_twait.c 2023-09-16

```

```

16:30:40.000000000 +0000
+++ ncurses-6.5-20241006/ncurses/tty/lib_twait.c 2024-10-05 22:47:12.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 1988-2018, 2023 Thomas E. Dickey                      *
+ * Copyright 1988-2023, 2024 Thomas E. Dickey                      *
* Copyright 1988-2015, 2016 Free Software Foundation, Inc.          *
*                                                                    *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -76,7 +76,7 @@
#endif
#undef CUR

-MODULE_ID("$Id: lib_twait.c,v 1.81 2023/09/16 16:30:40 tom Exp $")
+MODULE_ID("$Id: lib_twait.c,v 1.82 2024/10/05 22:47:12 tom Exp $")

/*
* Returns an elapsed time, in milliseconds (if possible).
@@ -168,7 +168,7 @@
* descriptors.
*/
NCURSES_EXPORT(int)
_nc_timed_wait(SCREEN *sp MAYBE_UNUSED,
+_nc_timed_wait(const
SCREEN *sp MAYBE_UNUSED,
    int mode MAYBE_UNUSED,
    int milliseconds,
    int *timeleft
Index: package/debian-mingw/changelog
--- ncurses-6.5-20240928+/package/debian-mingw/changelog 2024-09-28 09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/package/debian-mingw/changelog 2024-10-06 08:53:45.000000000 +0000
@@ -1,8 +1,8 @@
-ncurses6td (6.5+20240928) unstable; urgency=low
+ncurses6td (6.5+20241006) unstable; urgency=low

* latest weekly patch

- -- Thomas E. Dickey <dickey@invisible-island.net> Sat, 28 Sep 2024 05:53:21 -0400
+ -- Thomas E. Dickey <dickey@invisible-island.net> Sun, 06 Oct 2024 04:53:45 -0400

ncurses6 (5.9+20131005) unstable; urgency=low

Index: package/debian-mingw64/changelog
--- ncurses-6.5-20240928+/package/debian-mingw64/changelog 2024-09-28 09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/package/debian-mingw64/changelog 2024-10-06 08:53:45.000000000 +0000
@@ -1,8 +1,8 @@
-ncurses6td (6.5+20240928) unstable; urgency=low
+ncurses6td

```

(6.5+20241006) unstable; urgency=low

* latest weekly patch

- -- Thomas E. Dickey <dickey@invisible-island.net> Sat, 28 Sep 2024 05:53:21 -0400
+ -- Thomas E. Dickey <dickey@invisible-island.net> Sun, 06 Oct 2024 04:53:45 -0400

ncurses6 (5.9+20131005) unstable; urgency=low

Index: package/debian/changelog

--- ncurses-6.5-20240928+/package/debian/changelog 2024-09-28 09:53:21.000000000 +0000

+++ ncurses-6.5-20241006/package/debian/changelog 2024-10-06 08:53:45.000000000 +0000

@@ -1,8 +1,8 @@

-ncurses6td (6.5+20240928) unstable; urgency=low

+ncurses6td (6.5+20241006) unstable; urgency=low

* latest weekly patch

- -- Thomas E. Dickey <dickey@invisible-island.net> Sat, 28 Sep 2024 05:53:21 -0400
+ -- Thomas E. Dickey <dickey@invisible-island.net> Sun, 06 Oct 2024 04:53:45 -0400

ncurses6 (5.9+20120608) unstable; urgency=low

Index: package/mingw-ncurses.nsi

Prereq: 1.669

--- ncurses-6.5-20240928+/package/mingw-ncurses.nsi 2024-09-28 09:53:21.000000000 +0000

+++

ncurses-6.5-20241006/package/mingw-ncurses.nsi 2024-10-06 08:53:45.000000000 +0000

@@ -1,4 +1,4 @@

-\$Id: mingw-ncurses.nsi,v 1.669 2024/09/28 09:53:21 tom Exp \$

+\$Id: mingw-ncurses.nsi,v 1.671 2024/10/06 08:53:45 tom Exp \$

; TODO add examples

; TODO bump ABI to 6

@@ -10,7 +10,7 @@

!define VERSION_MAJOR "6"

!define VERSION_MINOR "5"

!define VERSION_YYYY "2024"

!define VERSION_MMDD "0928"

+!define VERSION_MMDD "1006"

!define VERSION_PATCH \${VERSION_YYYY}\${VERSION_MMDD}

!define MY_ABI "5"

Index: package/mingw-ncurses.spec

--- ncurses-6.5-20240928+/package/mingw-ncurses.spec 2024-09-28 09:53:21.000000000 +0000

+++ ncurses-6.5-20241006/package/mingw-ncurses.spec 2024-10-06 08:53:45.000000000 +0000

@@ -3,7 +3,7 @@

Summary: shared libraries for terminal handling

Name: mingw32-ncurses6
Version: 6.5
-Release: 20240928
+Release: 20241006
License: X11
Group: Development/Libraries
URL: <https://invisible-island.net/ncurses/>
Index: package/ncurses.spec
--- ncurses-6.5-20240928+/package/ncurses.spec 2024-09-28
09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/package/ncurses.spec 2024-10-06 08:53:45.000000000 +0000
@@ -1,7 +1,7 @@

Summary: shared libraries for terminal handling

Name: ncurses6
Version: 6.5
-Release: 20240928
+Release: 20241006
License: X11
Group: Development/Libraries
URL: <https://invisible-island.net/ncurses/>
Index: package/ncurset.spec
--- ncurses-6.5-20240928+/package/ncurset.spec 2024-09-28 09:53:21.000000000 +0000
+++ ncurses-6.5-20241006/package/ncurset.spec 2024-10-06 08:53:45.000000000 +0000
@@ -1,7 +1,7 @@

Summary: Curses library with POSIX thread support.

Name: ncurset6
Version: 6.5
-Release: 20240928
+Release: 20241006
License: X11
Group: Development/Libraries
Source: ncurses-%{version}-%{release}.tgz
Index: test/blue.c
Prereq: 1.55
--- ncurses-6.5-20240928+/test/blue.c 2022-12-10 23:31:31.000000000 +0000
+++ ncurses-6.5-20241006/test/blue.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@

/*****

-

* Copyright 2019-2021,2022 Thomas E. Dickey *

+ * Copyright 2019-2022,2024 Thomas E. Dickey *

* Copyright 1998-2016,2017 Free Software Foundation, Inc. *

* *

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -37,7 +37,7 @@

*****/

/*

```
- * $Id: blue.c,v 1.55 2022/12/10 23:31:31 tom Exp $
+ * $Id: blue.c,v 1.56 2024/10/05 19:26:24 tom Exp $
*/
```

```
#include <test.priv.h>
@@ -437,7 +437,7 @@
static void
use_pc_display(void)
{
- char *check = nl_langinfo(CODESET);
+ const char *check = nl_langinfo(CODESET);
    if (!strcmp(check, "UTF-8")) {
#if USE_WIDEC_SUPPORT
    suits = uglyphs;
@@ -448,8 +448,8 @@
        !strcmp(check, "CP437") ||
        !strcmp(check,
"IBM850") ||
        !strcmp(check, "CP850")) {
- char *smacs = tigetstr("smacs");
- char *smpch = tigetstr("smpch");
+ const char *smacs = tigetstr("smacs");
+ const char *smpch = tigetstr("smpch");
    /*
     * The ncurses library makes this check to decide whether to allow
     * the alternate character set for the (normally) nonprinting codes.
```

Index: test/bs.c

Prereq: 1.79

--- ncurses-6.5-20240928+/test/bs.c 2023-05-27 20:13:10.000000000 +0000

+++ ncurses-6.5-20241006/test/bs.c 2024-10-05 19:26:24.000000000 +0000

@@ -1,5 +1,5 @@

/*****

```
- * Copyright 2018-2022,2023 Thomas E. Dickey *
+ * Copyright 2018-2023,2024 Thomas E. Dickey *
* Copyright 1998-2016,2017 Free Software Foundation, Inc. *
* * *
```

```
* Permission is hereby granted, free of charge,
to any person obtaining a *
```

@@ -35,7 +35,7 @@

```
* v2.0 featuring strict ANSI/POSIX conformance, November 1993.
```

```
* v2.1 with ncurses mouse support, September 1995
```

```
*
```

```
- * $Id: bs.c,v 1.79 2023/05/27 20:13:10 tom Exp $
```

```
+ * $Id: bs.c,v 1.80 2024/10/05 19:26:24 tom Exp $
```

```
*/
```

```
#include <test.priv.h>
```

@@ -147,7 +147,7 @@

```

    bool placed; /* has it been placed on the board? */
} ship_t;

-static bool checkplace(int b, ship_t * ss, int vis);
+static bool checkplace(int b, const ship_t * ss, int vis);

#define SHIPIT(name, symbol, length) { name, 0, symbol, length, 0,0, 0, FALSE }

@@ -686,7 +686,7 @@
}

static bool
-checkplace(int b, ship_t * ss, int vis)
+checkplace(int b, const ship_t * ss, int vis)
{
    int l, xend, yend;

Index: test/cardfile.c
Prereq: 1.51
--- ncurses-6.5-20240928+/test/cardfile.c 2022-12-04 00:40:11.000000000 +0000
+++ ncurses-6.5-20241006/test/cardfile.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
-
+ * Copyright 2019-2021,2022 Thomas E. Dickey *
+ * Copyright 2019-2022,2024 Thomas E. Dickey *
+ * Copyright 1999-2016,2017 Free Software Foundation, Inc. *
+ *
+ * Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
/*
+ * Author: Thomas E. Dickey
+ *
- * $Id: cardfile.c,v 1.51 2022/12/04 00:40:11 tom Exp $
+ * $Id: cardfile.c,v 1.52 2024/10/05 19:26:24 tom Exp $
+ *
+ * File format: text beginning in column 1 is a title; other text is content.
+ */
@@ -161,7 +161,7 @@
}

static CARD *
-find_card(char *title)
+find_card(const char *title)
{
    CARD *card;

@@ -173,7 +173,7 @@

```

```

}

static void
-read_data(char *fname)
+read_data(const char *fname)
{
    FILE *fp;

@@ -341,7 +341,7 @@
}

static FIELD **
-make_fields(CARD * p, int form_high,
int form_wide)
+make_fields(const CARD * p, int form_high, int form_wide)
{
    FIELD **f = typeCalloc(FIELD *, (size_t) 3);

@@ -385,7 +385,7 @@
/*****

static void
-cardfile(char *fname)
+cardfile(const char *fname)
{
    WINDOW *win;
    CARD *p;
Index: test/chgat.c
Prereq: 1.21
--- ncurses-6.5-20240928+/test/chgat.c 2024-06-29 17:45:32.000000000 +0000
+++ ncurses-6.5-20241006/test/chgat.c 2024-10-06 22:35:55.000000000 +0000
@@ -27,7 +27,7 @@
* authorization.
*
*****/
/*
- * $Id: chgat.c,v 1.21 2024/06/29 17:45:32 tom Exp $
+ * $Id: chgat.c,v 1.23 2024/10/06 22:35:55 tom Exp $
*
* test-driver for chgat/wchgat/mvchgat/mvwchgat
*/
@@ -160,7 +160,7 @@
}

static void
-do_subwindow(WINDOW *win, STATUS * sp, void func(WINDOW *))
+do_subwindow(WINDOW *win, const STATUS
* sp, void func(WINDOW *))
{

```

```

WINDOW *win1 = newwin(sp->y_max - 2, sp->x_max - 2,
    sp->y_beg + 1, sp->x_beg + 1);
@@ -204,7 +204,7 @@
static void
show_help(WINDOW *win)
{
- static const char *msgs[] =
+ static NCURSES_CONST char *msgs[] =
    {
        "Basic commands:"
        ,"Use h/j/k/l or arrow keys to move the cursor."
Index: test/clip_printw.c
Prereq: 1.23
--- ncurses-6.5-20240928+/test/clip_printw.c 2023-09-30 19:57:44.000000000 +0000
+++ ncurses-6.5-20241006/test/clip_printw.c 2024-10-06 22:36:25.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2019-2022,2023 Thomas E. Dickey                *
+ * Copyright 2019-2023,2024 Thomas E. Dickey                *
 * Copyright 2008-2016,2017 Free Software Foundation, Inc.    *
 *                                                              *
 * Permission is hereby granted, free of charge,
to any person obtaining a *
@@ -27,7 +27,7 @@
 * authorization.                                           *
 *****/
/*
- * $Id: clip_printw.c,v 1.23 2023/09/30 19:57:44 tom Exp $
+ * $Id: clip_printw.c,v 1.25 2024/10/06 22:36:25 tom Exp $
 *
 * demonstrate how to use printw with/without wrapping.
 */
@@ -212,7 +212,7 @@
}

static void
-do_subwindow(WINDOW *win, STATUS *sp, void func(WINDOW *))
+do_subwindow(WINDOW *win, const STATUS *sp, void func(WINDOW *))
{
    WINDOW *win1 = newwin(sp->y_max - (2 * MARGIN),
        sp->x_max - (2 * MARGIN),
@@ -268,7 +268,7 @@
static void
show_help(WINDOW *win)
{
- static const char *msgs[] =
+ static NCURSES_CONST char *msgs[] =
    {

```

```

"Basic commands:"
,"Use h/j/k/l or arrow keys to move the cursor."
Index: test/color_name.h
Prereq: 1.9
--- ncurses-6.5-20240928+/test/color_name.h 2021-04-24 23:25:29.000000000 +0000
+++ ncurses-6.5-20241006/test/color_name.h 2024-10-05
19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2018-2020,2021 Thomas E. Dickey                *
+ * Copyright 2018-2021,2024 Thomas E. Dickey                *
 * Copyright 2011-2012,2016 Free Software Foundation, Inc.    *
 *                                                              *
 * Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
 * authorization.                                           *
*****/

/*
- * $Id: color_name.h,v 1.9 2021/04/24 23:25:29 tom Exp $
+ * $Id: color_name.h,v 1.10 2024/10/05 19:26:24 tom Exp $
 */

#ifndef __COLORNAME_H
@@ -63,7 +63,6 @@
{
    int result = 0;
    char *endp = 0;
-   size_t n;

    if ((result = (int) strtol(color,
&endp, 0)) >= 0
&& (endp == 0 || *endp == 0)) {
@@ -71,6 +70,7 @@
    } else if (!strcmp(color, "default")) {
        result = -1;
    } else {
+   size_t n;
    for (n = 0; n < SIZEOF(the_color_names); ++n) {
        if (!strcmp(the_color_names[n], color)) {
            result = (int) n;

```

```

Index: test/demo_defkey.c
Prereq: 1.35
--- ncurses-6.5-20240928+/test/demo_defkey.c 2024-01-20 20:46:12.000000000 +0000
+++ ncurses-6.5-20241006/test/demo_defkey.c 2024-10-05 18:20:04.000000000 +0000
@@ -27,7 +27,7 @@
 * authorization.                                           *
*****/

/*

```

```

- * $Id: demo_defkey.c,v 1.35 2024/01/20 20:46:12 tom Exp $
+ * $Id: demo_defkey.c,v 1.36 2024/10/05 18:20:04 tom Exp $
*
* Demonstrate the define_key() function.
* Thomas Dickey - 2002/11/23
@@ -199,7 +199,7 @@
}

```

```

static void
-redefine(WINDOW *win, char *string, int code)
+redefine(WINDOW *win, const char *string, int code)
{

```

```

    really_define_key(win, string, code);
}

```

Index: test/demo_forms.c

Prereq: 1.63

--- ncurses-6.5-20240928+/test/demo_forms.c 2023-11-11 00:29:53.000000000 +0000

+++ ncurses-6.5-20241006/test/demo_forms.c 2024-10-06 21:15:12.000000000 +0000

@@ -1,5 +1,5 @@

/******

```

- * Copyright 1988-2022,2023 Thomas E. Dickey          *
+ * Copyright 1988-2023,2024 Thomas E. Dickey          *
* Copyright 2003-2016,2017 Free Software Foundation, Inc. *
*
*

```

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -27,7 +27,7 @@

* authorization. *

*****/

/*

- * \$Id: demo_forms.c,v 1.63 2023/11/11 00:29:53 tom Exp \$

+ * \$Id: demo_forms.c,v 1.67

2024/10/06 21:15:12 tom Exp \$

*

* Demonstrate a variety of functions from the form library.

* Thomas Dickey - 2003/4/26

@@ -148,7 +148,7 @@

}

static FIELD *

-make_label(const char *label, int frow, int fcol)

+make_label(NCURSES_CONST char *label, int frow, int fcol)

{

FIELD *f = new_field(1, (int) strlen(label), frow, fcol, 0, 0);

@@ -242,7 +242,7 @@

#define O_SELECTABLE (O_ACTIVE | O_VISIBLE)

```

static FIELD *
-another_field(FORM *form, const FIELD *const field)
+another_field(NCURSES_CONST FORM *form, NCURSES_CONST FIELD *const field)
{
    FIELD **f = form_fields(form);
    FIELD *result = 0;
@@ -306,9 +306,9 @@
}

static void
-show_current_field(WINDOW *win, FORM *form)
+show_current_field(WINDOW *win, NCURSES_CONST FORM *form)
{
-    FIELD *field;
+    NCURSES_CONST FIELD *field;
    int field_rows, field_cols, field_max;
    int currow, curcol;

@@ -325,7 +325,7 @@
    waddch(win, '\n');

    if ((field = current_field(form)) != 0) {
-    FIELDTYPE *type;
+    NCURSES_CONST
    FIELDTYPE *type;
    int nbuf;

    wprintw(win, "Page %d%s, Field %d/%d%s:",
@@ -383,7 +383,7 @@

    waddstr(win, "\n");
    for (nbuf = 0; nbuf <= 2; ++nbuf) {
-    char *buffer;
+    NCURSES_CONST char *buffer;
        if ((buffer = field_buffer(field, nbuf)) != 0) {
            wprintw(win, "buffer %d:", nbuf);
            (void) wattrset(win, A_REVERSE);
@@ -408,8 +408,8 @@
            int c;
            unsigned n = 0;
            int pg;
-            const char *fname;
-            static const char *my_enum[] =
+            NCURSES_CONST char *fname;
+            static NCURSES_CONST char *my_enum[] =
                {"first", "second", "third", 0};

#ifdef NCURSES_MOUSE_VERSION

```


@@ -515,7 +515,7 @@

f[n] = (FIELD *) 0;

```
    if ((form = new_form(f)) != 0) {
- WINDOW *w;
+ NCURSES_CONST WINDOW *w;
    WINDOW *also;
    int finished = 0;
```

Index: test/demo_menus.c

Prereq: 1.80

--- ncurses-6.5-20240928+/test/demo_menus.c 2023-05-27 20:13:10.000000000 +0000

+++ ncurses-6.5-20241006/test/demo_menus.c 2024-10-06 19:58:28.000000000 +0000

@@ -1,5

+1,5 @@

/*****

```
- * Copyright 1919-2022,2023 Thomas E. Dickey          *
+ * Copyright 1919-2023,2024 Thomas E. Dickey          *
* Copyright 2003-2016,2017 Free Software Foundation, Inc. *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
```

@@ -27,7 +27,7 @@

```
* authorization.                                *
```

*****/

/*

```
- * $Id: demo_menus.c,v 1.80 2023/05/27 20:13:10 tom Exp $
```

```
+ * $Id: demo_menus.c,v 1.82 2024/10/06 19:58:28 tom Exp $
```

```
*
```

```
* Demonstrate a variety of functions from the menu library.
```

```
* Thomas Dickey - 2005/4/9
```

@@ -185,7 +185,7 @@

}

static int

```
-menu_getc(MENU * m)
```

```
+menu_getc(NCURSES_CONST MENU * m)
```

```
{
    return wGetchar(menu_win(m));
}
```

@@ -214,7 +214,7 @@

```
my_menu_init(MENU * menu)
```

```
{
    Trace(("called MenuHook my_menu_init"));
- mvwprintw(status, 2, 0, "menu_init %p", (void *) menu);
+ mvwprintw(status, 2, 0, "menu_init %p", (const void *) menu);
    wclrtoeol(status);
    wrefresh(status);
}
```

```

@@ -223,7 +223,7 @@
my_menu_term(MENU * menu)
{
    Trace(("called MenuHook my_menu_term"));
-   mvwprintw(status, 2, 0, "menu_term %p", (void *) menu);
+   mvwprintw(status, 2, 0, "menu_term %p", (const void *) menu);
    wclrtoeol(status);
    wrefresh(status);
}
@@ -231,7 +231,7 @@
static void
my_item_init(MENU * menu)
{
-   ITEM *item = current_item(menu);
+   NCURSES_CONST ITEM *item = current_item(menu);
    const char *name = item_name(item);

    Trace(("called MenuHook my_item_init (%s)", name));
@@ -243,7 +243,7 @@
static void
my_item_term(MENU * menu)
{
-   ITEM *item = current_item(menu);
+   NCURSES_CONST ITEM *item = current_item(menu);
    const char *name = item_name(item);

    Trace(("called MenuHook my_item_term (%s)", name));
@@ -348,7 +348,7 @@

/* force the given menu to appear */
static void
-menu_display(MENU * m)
+menu_display(NCURSES_CONST MENU * m)
{
    touchwin(menu_win(m));
    wrefresh(menu_win(m));
@@ -584,10 +584,10 @@
* the others
*/

static bool
-update_trace_menu(MENU * m)
+update_trace_menu(NCURSES_CONST MENU * m)
{
    ITEM **items;
-   ITEM *i;
+   NCURSES_CONST ITEM *i;
    bool changed = FALSE;

```

```

    items = menu_items(m);
@@ -613,7 +613,7 @@
    int result;

    for (ip = menu_items(mpTrace); *ip; ip++) {
- MENU_DATA *td = (MENU_DATA *) item_userptr(*ip);
+ NCURSES_CONST MENU_DATA *td = (MENU_DATA *) item_userptr(*ip);
    unsigned mask = td->mask;
    if (mask == 0)
        set_item_value(*ip, _nc_tracing == 0);
@@ -628,7 +628,7 @@
    unsigned newtrace = 0;
    for (ip = menu_items(mpTrace); *ip; ip++) {
    if (item_value(*ip)) {
- MENU_DATA *td = (MENU_DATA *) item_userptr(*ip);
+ NCURSES_CONST
MENU_DATA *td = (MENU_DATA *) item_userptr(*ip);
        newtrace |= td->mask;
    }
    }
@@ -718,7 +718,9 @@
}

static int
-move_menu(MENU * menu, MENU * current, int by_y, int by_x)
+move_menu(NCURSES_CONST MENU * menu,
+ NCURSES_CONST MENU * current,
+ int by_y, int by_x)
{
    WINDOW *top_win = menu_win(menu);
    WINDOW *sub_win = menu_sub(menu);
@@ -798,7 +800,7 @@
#endif /* defined(KEY_RESIZE) && NCURSES_EXT_FUNCS */

static void
-show_status(int ch, MENU * menu)
+show_status(int ch, NCURSES_CONST MENU * menu)
{
    wmove(status, 0, 0);
    wprintw(status, "key %s, menu %d, mark %s, match %s",
@@ -919,7 +921,7 @@
wrefresh(menu_win(last_menu));
if (code == E_UNKNOWN_COMMAND
    || code == E_NOT_POSTED) {
- ITEM *item = current_item(last_menu);
+ NCURSES_CONST ITEM *item = current_item(last_menu);
    MENU_DATA *td = (MENU_DATA *) item_userptr(item);
    td->func((int) td->mask);

```

```

}
Index: test/demo_new_pair.c
Prereq: 1.28
--- ncurses-6.5-20240928+/test/demo_new_pair.c 2024-06-29
19:30:26.000000000 +0000
+++ ncurses-6.5-20241006/test/demo_new_pair.c 2024-10-05 18:21:44.000000000 +0000
@@ -27,7 +27,7 @@
* authorization.
*
*****
/*
- * $Id: demo_new_pair.c,v 1.28 2024/06/29 19:30:26 tom Exp $
+ * $Id: demo_new_pair.c,v 1.29 2024/10/05 18:21:44 tom Exp $
*
* Demonstrate the alloc_pair() function.
*/
@@ -49,7 +49,7 @@
static bool
valid_cap(NCURSES_CONST char *name)
{
- char *value = tigetstr(name);
+ const char *value = tigetstr(name);
return (value != 0 && value != (char *) -1) ? TRUE : FALSE;
}

Index: test/demo_panels.c
Prereq: 1.48
--- ncurses-6.5-20240928+/test/demo_panels.c 2022-12-04 00:40:11.000000000 +0000
+++ ncurses-6.5-20241006/test/demo_panels.c 2024-10-06 20:01:21.000000000 +0000
@@ -1,5 +1,5 @@
/*****
-
* Copyright 2018-2020,2022 Thomas E. Dickey
+ * Copyright 2018-2022,2024 Thomas E. Dickey
* Copyright 2003-2016,2017 Free Software Foundation, Inc.
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -27,7 +27,7 @@
* authorization.
*
*****
/*
- * $Id: demo_panels.c,v 1.48 2022/12/04 00:40:11 tom Exp $
+ * $Id: demo_panels.c,v 1.50 2024/10/06 20:01:21 tom Exp $
*
* Demonstrate a variety of functions from the panel library.
*/
@@ -42,7 +42,7 @@
#define TEMP_POS '>'

```

```

typedef void (*InitPanel) (void);
-typedef void (*FillPanel) (PANEL *);
+typedef void (*FillPanel) (NCURSES_CONST PANEL *);

static bool use_colors = FALSE;
static bool
    unboxed = FALSE;
@@ -301,7 +301,7 @@
int code;
int y0, x0;
int y1, x1;
- WINDOW *win = panel_window(pans[which]);
+ NCURSES_CONST WINDOW *win = panel_window(pans[which]);
char also[80];

getbegyx(win, y0, x0);
@@ -368,7 +368,7 @@
}

static void
-fill_panel(PANEL *pan)
+fill_panel(NCURSES_CONST PANEL *pan)
{
    WINDOW *win = panel_window(pan);
    const char *userptr = (const char *) panel_userptr(pan);
@@ -388,7 +388,7 @@
}

static void
-fill_unboxed(PANEL *pan)
+fill_unboxed(NCURSES_CONST PANEL *pan)
{
    WINDOW *win = panel_window(pan);
    const char *userptr = (const char *) panel_userptr(pan);
@@ -431,7 +431,7 @@
}

static void
-fill_wide_panel(PANEL *pan)
+fill_wide_panel(NCURSES_CONST PANEL *pan)
{
    WINDOW *win = panel_window(pan);
    int num = ((const char *) panel_userptr(pan))[1];
@@ -453,7 +453,7 @@
#define MAX_PANELS 5

static int
-which_panel(PANEL *px[MAX_PANELS + 1], PANEL *pan)

```

```

+which_panel(PANEL *px[MAX_PANELS + 1], NCURSES_CONST
PANEL *pan)
{
    int result = 0;
    int j;
Index: test/demo_termcap.c
Prereq: 1.65
--- ncurses-6.5-20240928+/test/demo_termcap.c 2023-05-27 20:13:10.000000000 +0000
+++ ncurses-6.5-20241006/test/demo_termcap.c 2024-10-06 20:05:09.000000000 +0000
@@ -1,5 +1,5 @@
/******
- * Copyright 2019-2022,2023 Thomas E. Dickey *
+ * Copyright 2019-2023,2024 Thomas E. Dickey *
* Copyright 2005-2016,2017 Free Software Foundation, Inc. *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
/*
* Author: Thomas E. Dickey
*
- * $Id: demo_termcap.c,v 1.65 2023/05/27 20:13:10 tom Exp $
+ * $Id: demo_termcap.c,v 1.70 2024/10/06 20:05:09 tom Exp $
*
* A simple demo of the termcap interface.
*/
@@ -256,7 +256,7 @@

* to hold all of the strings extracted from the terminal entry.
*/
char area[1024], *ap = area;
- char *str;
+ NCURSES_CONST char *str;
int num;

if ((str = tgetstr(cap, &ap)) != 0) {
@@ -289,7 +289,7 @@
}

static void
-brute_force(const char *name)
+brute_force(NCURSES_CONST char *name)
{
    char buffer[1024];

@@ -369,7 +369,7 @@
#ifdef NCURSES_VERSION
if (x_opt && (my_blob == 0) && y_opt) {
#endif
#ifdef NCURSES_XNAMES

```

```

-   TERMTYPE *term = (TERMTYPE *) cur_term;
+   NCURSES_CONST TERMTYPE *term = (TERMTYPE *) cur_term;
    if (term != 0
        && ((NUM_BOOLEANS(term) != BOOLCOUNT)
            || (NUM_NUMBERS(term) != NUMCOUNT)
@@@ -781,12 +781,12 @@@
{
    int ch;
    int n;
-   char *name;
+   NCURSES_CONST char *name;
    bool a_opt = FALSE;
#if defined(NCURSES_VERSION) || defined(HAVE_CURSES_DATA_OSPEED)
    bool v_opt = FALSE;
#endif
-   char *input_name = 0;
+   NCURSES_CONST char *input_name = 0;

    int repeat;
    int r_opt = 1;
@@@ -867,7
+867,7 @@@
    } else if ((name = getenv("TERM")) != 0) {
        brute_force(name);
    } else {
-   static char dumb[] = "dumb";
+   static NCURSES_CONST char dumb[] = "dumb";
        brute_force(dumb);
    }
}
@@@ -894,7 +894,7 @@@
    } else if ((name = getenv("TERM")) != 0) {
        demo_termcap(name);
    } else {
-   static char dumb[] = "dumb";
+   static NCURSES_CONST char dumb[] = "dumb";
        demo_termcap(dumb);
    }
}
Index: test/demo_terminfo.c
Prereq: 1.57
--- ncurses-6.5-20240928+/test/demo_terminfo.c 2023-05-27 20:13:10.000000000 +0000
+++ ncurses-6.5-20241006/test/demo_terminfo.c 2024-10-06 21:16:43.000000000 +0000
@@@ -1,5 +1,5 @@@
/*****
- * Copyright 2019-2022,2023 Thomas E. Dickey          *
+ * Copyright 2019-2023,2024 Thomas E. Dickey          *
* Copyright 2009-2016,2017 Free Software Foundation, Inc.      *

```

```

*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
/*
* Author: Thomas E. Dickey
*
- * $Id: demo_terminfo.c,v 1.57 2023/05/27 20:13:10 tom Exp $
+ * $Id: demo_terminfo.c,v 1.59 2024/10/06 21:16:43 tom Exp $
*
* A simple demo of the terminfo interface.
*/
@@ -253,7 +253,7 @@
#define LegalItem(c,n) (n)

static void
-brute_force(const char *name)
+brute_force(NCURSES_CONST char *name)
{
#define MAX_FORCE 5 /* omit "colors", since CPU-time is a problem */
    static const char legal[] = "\
@@ -324,7 +324,7 @@
#endif

static void
-demo_terminfo(char *name)
+demo_terminfo(NCURSES_CONST char *name)
{
    unsigned n;
    NCURSES_CONST char *cap;
@@ -366,7 +366,7 @@
    if (x_opt && (my_blob == 0)) {
    if (y_opt) {
#ifdef NCURSES_XNAMES
-    TERMTYPE *term = (TERMTYPE *) cur_term;
+    const TERMTYPE *term = (TERMTYPE *) cur_term;
    if (term != 0
        && ((NUM_BOOLEANS(term)
            != BOOLCOUNT)
            || (NUM_NUMBERS(term) != NUMCOUNT)
@@ -813,9 +813,9 @@
    int ch;
    int n;
    int repeat;
-    char *name;
+    NCURSES_CONST char *name;
    int r_opt = 1;
-    char *input_name = 0;

```



```

+   const char *input_name = 0;

    while ((ch = getopt(argc, argv, OPTS_COMMON "abd:e:fi:nqr:sxy")) != -1) {
switch (ch) {
@@ -891,7 +891,7 @@
    } else if ((name = getenv("TERM")) != 0) {
        brute_force(name);
    } else {
-   static char dumb[] = "dumb";
+   static NCURSES_CONST char dumb[] = "dumb";
        brute_force(dumb);
    }
}
@@ -918,7 +918,7 @@
    } else if ((name = getenv("TERM")) != 0) {
        demo_terminfo(name);
    } else {
-   static char dumb[] = "dumb";
+   static NCURSES_CONST char dumb[] = "dumb";
        demo_terminfo(dumb);
    }
}
Index: test/dots.c
Prereq: 1.45
--- ncurses-6.5-20240928+/test/dots.c 2023-01-07 17:21:48.000000000 +0000
+++ ncurses-6.5-20241006/test/dots.c 2024-10-06 21:17:54.000000000 +0000
@@ -1,5 +1,5 @@
/*****
-
* Copyright 2019-2022,2023 Thomas E. Dickey                *
+ * Copyright 2019-2023,2024 Thomas E. Dickey                *
* Copyright 1999-2013,2017 Free Software Foundation, Inc.    *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
/*
* Author: Thomas E. Dickey <dickey@clark.net> 1999
*
- * $Id: dots.c,v 1.45 2023/01/07 17:21:48 tom Exp $
+ * $Id: dots.c,v 1.46 2024/10/06 21:17:54 tom Exp $
*
* A simple demo of the terminfo interface.
*/
@@ -61,7 +61,7 @@
}

static bool
-outs(const char *s)

```

```

+outs(NCURSES_CONST char *s)
{
    if (VALID_STRING(s)) {
        tputs(s, 1, outc);
Index: test/dots_mvcur.c
Prereq: 1.31
--- ncurses-6.5-20240928+/test/dots_mvcur.c 2023-01-07 17:21:48.000000000 +0000
+++ ncurses-6.5-20241006/test/dots_mvcur.c 2024-10-06
21:17:54.000000000 +0000
@@@ -1,5 +1,5 @@@
/******
- * Copyright 2019-2022,2023 Thomas E. Dickey                *
+ * Copyright 2019-2023,2024 Thomas E. Dickey                *
* Copyright 2007-2013,2017 Free Software Foundation, Inc.    *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@@ -30,7 +30,7 @@@
/*
* Author: Thomas E. Dickey - 2007
*
- * $Id: dots_mvcur.c,v 1.31 2023/01/07 17:21:48 tom Exp $
+ * $Id: dots_mvcur.c,v 1.32 2024/10/06 21:17:54 tom Exp $
*
* A simple demo of the terminfo interface, and mvcur.
*/
@@@ -62,7 +62,7 @@@
}

```

```

static bool
-outs(const char *s)
+outs(NCURSES_CONST char *s)
{
    if (VALID_STRING(s)) {
        tputs(s, 1, outc);
Index: test/dots_termcap.c
Prereq: 1.33
--- ncurses-6.5-20240928+/test/dots_termcap.c 2024-08-31
15:55:46.000000000 +0000
+++ ncurses-6.5-20241006/test/dots_termcap.c 2024-10-06 21:18:35.000000000 +0000
@@@ -30,7 +30,7 @@@
/*
* Author: Thomas E. Dickey
*
- * $Id: dots_termcap.c,v 1.33 2024/08/31 15:55:46 tom Exp $
+ * $Id: dots_termcap.c,v 1.35 2024/10/06 21:18:35 tom Exp $
*
* A simple demo of the termcap interface.
*/

```

```
@@ -113,7 +113,7 @@  
}
```

```
static bool  
-outs(char *s)  
+outs(NCURSES_CONST char *s)  
{  
    if (VALID_STRING(s)) {  
        tputs(s, 1, outc);  
@@ -178,7 +178,7 @@  
get_number(NCURSES_CONST char *cap, const char *env)  
{  
    int result = tgetnum(cap);  
-   char *value = env ? getenv(env) : 0;  
+   const char *value = env ? getenv(env) : 0;  
    if (value != 0 && *value != 0) {  
        char *next = 0;  
        long check = strtol(value, &next, 10);  
@@ -229,7 +229,7 @@  
        double c;  
        char buffer[1024];  
        char area[1024];  
-   char *name;  
+   NCURSES_CONST char *name;  
        size_t need;  
        char *my_env;
```

Index: test/dup_field.c

Prereq: 1.9

--- ncurses-6.5-20240928+/test/dup_field.c 2024-06-29

17:45:32.000000000 +0000

+++ ncurses-6.5-20241006/test/dup_field.c 2024-10-06 20:07:19.000000000 +0000

```
@@ -26,7 +26,7 @@
```

```
* authorization.
```

```
*
```

```
*****/
```

```
/*
```

```
- * $Id: dup_field.c,v 1.9 2024/06/29 17:45:32 tom Exp $
```

```
+ * $Id: dup_field.c,v 1.11 2024/10/06 20:07:19 tom Exp $
```

```
*
```

```
* Demonstrate dup_field().
```

```
*/
```

```
@@ -146,27 +146,27 @@
```

```
}
```

```
static FieldAttrs *
```

```
-my_field_attrs(FIELD *f)
```

```
+my_field_attrs(NCURSES_CONST FIELD *f)
```

```
{
```

```

    return (FieldAttrs *) field_userptr(f);
}

static int
-buffer_length(FIELD *f)
+buffer_length(NCURSES_CONST FIELD *f)
{
    return my_field_attrs(f)->row_lengths[0];
}

static void
-set_buffer_length(FIELD *f, int length)
+set_buffer_length(NCURSES_CONST FIELD *f, int length)
{
    my_field_attrs(f)->row_lengths[0] = length;
}

static int
-offset_in_field(FORM *form)
+offset_in_field(NCURSES_CONST
FORM *form)
{
-   FIELD *field = current_field(form);
+   NCURSES_CONST FIELD *field = current_field(form);
    int currow, curcol;

    form_getyx(form, currow, curcol);
@@ -245,7 +245,7 @@

default:
    modified = (ch < MIN_FORM_COMMAND
-   && isprint(ch));
+   && isprint(UChar(ch)));
    break;
}

Index: test/filter.c
Prereq: 1.38
--- ncurses-6.5-20240928+/test/filter.c 2022-12-04 00:40:11.000000000 +0000
+++ ncurses-6.5-20241006/test/filter.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2019-2020,2022 Thomas E. Dickey                *
+ * Copyright 2019-2022,2024 Thomas E. Dickey                *
 * Copyright 1998-2016,2017 Free Software Foundation, Inc.    *
 *                                                              *
 * Permission is hereby granted, free of charge, to any person obtaining a *
@@

```

```

-30,7 +30,7 @@
/*
 * Author: Thomas E. Dickey 1998
 *
- * $Id: filter.c,v 1.38 2022/12/04 00:40:11 tom Exp $
+ * $Id: filter.c,v 1.39 2024/10/05 19:26:24 tom Exp $
 *
 * An example of the 'filter()' function in ncurses, this program prompts
 * for commands and executes them (like a command shell). It illustrates
@@ -65,7 +65,7 @@
    if (clocked) {
    if (limit >= 3) {
        time_t now = time((time_t *) 0);
-    struct tm *my = localtime(&now);
+    const struct tm *my = localtime(&now);
        char buffer[80];
        int skip, y, x;
        int margin;
Index: test/form_driver_w.c
Prereq: 1.17
--- ncurses-6.5-20240928+/test/form_driver_w.c 2022-12-10 23:31:31.000000000 +0000
+++ ncurses-6.5-20241006/test/form_driver_w.c 2024-10-06 23:13:40.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey
+ * Copyright 2020-2022,2024 Thomas E. Dickey
 *
 * Copyright 2013-2014,2017 Free Software Foundation, Inc.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
@@ -32,7 +32,7 @@
*****/

/*
- * $Id: form_driver_w.c,v 1.17 2022/12/10 23:31:31 tom Exp $
+ * $Id: form_driver_w.c,v 1.18 2024/10/06 23:13:40 tom Exp $
 *
 * Test form_driver_w (int, int, wchar_t), a wide char aware
 * replacement of form_driver.
@@ -68,7 +68,7 @@
int
main(int argc, char *argv[])
{
-    static const char *help[] =
+    static NCURSES_CONST char *help[] =
    {
        "Commands:",
        " ^D,^Q,ESC      - quit program",

```

```

Index: test/gdc.c
Prereq: 1.60
--- ncurses-6.5-20240928+/test/gdc.c 2024-06-29 19:59:06.000000000 +0000
+++ ncurses-6.5-20241006/test/gdc.c 2024-10-06 20:08:59.000000000 +0000
@@ -34,7 +34,7 @@
* modified 10-18-89
for curses (jrl)
* 10-18-89 added signal handling
*
- * $Id: gdc.c,v 1.60 2024/06/29 19:59:06 tom Exp $
+ * $Id: gdc.c,v 1.62 2024/10/06 20:08:59 tom Exp $
*/

#include <test.priv.h>
@@ -181,7 +181,7 @@
    int check;
    time_t result;
    char c = 0;
-   struct tm *tm;
+   NCURSES_CONST struct tm *tm;

    switch (sscanf(value, "%d:%d:%d%c", &hh, &mm, &ss, &c)) {
    default:
@@ -243,7 +243,7 @@
main(int argc, char *argv[])
{
    time_t now;
-   struct tm *tm;
+   NCURSES_CONST struct tm *tm;
    long t, a;
    int i, j, s, k, ch;
    int count = 0;
@@ -293,7 +293,7 @@

    InitAndCatch({
    if (redirected) {
-   char *name = getenv("TERM");
+   NCURSES_CONST char *name = getenv("TERM");
        if (name == 0
            || newterm(name, ofp, ifp) == 0) {
            fprintf(stderr, "cannot open terminal\n");
@@ -431,7 +431,7 @@
        _nc_STRNCPY(buf, ctime(&now), (size_t) 30);
        {
            char *d2 = buf + 10;
-   char *s2 = buf + 19;
+   NCURSES_CONST char *s2 = buf + 19;

```

```

while ((*d2++ = *s2++) != '\0') ;
}
MvAddStr(16, 30, buf);
Index: test/inch_wide.c
Prereq: 1.13
--- ncurses-6.5-20240928+/test/inch_wide.c 2022-12-10 23:55:34.000000000 +0000
+++ ncurses-6.5-20241006/test/inch_wide.c 2024-10-06 23:14:31.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 1919-2020,2022 Thomas E. Dickey                *
+ * Copyright 1919-2022,2024 Thomas E. Dickey                *
 * Copyright 2007-2010,2017 Free Software Foundation, Inc.    *
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
@@ -27,7 +27,7 @@
 * authorization.
*****/

/*
- * $Id: inch_wide.c,v 1.13 2022/12/10 23:55:34 tom Exp $
+
 * $Id: inch_wide.c,v 1.14 2024/10/06 23:14:31 tom Exp $
 */
/*
    int in_wch(cchar_t *wcval);
@@ -61,7 +61,7 @@
static int
test_inchs(int level, char **argv, WINDOW *chrwin, WINDOW *strwin)
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
    "Test input from screen using inch(), etc., in a moveable viewport.",
    ""
}
Index: test/inchs.c
Prereq: 1.19
--- ncurses-6.5-20240928+/test/inchs.c 2022-12-11 00:01:39.000000000 +0000
+++ ncurses-6.5-20241006/test/inchs.c 2024-10-06 21:17:54.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 1919-2020,2022 Thomas E. Dickey                *
+ * Copyright 1919-2022,2024 Thomas E. Dickey                *
 * Copyright 2007-2012,2017 Free Software Foundation, Inc.    *
 *
 * Permission is hereby granted, free of
charge, to any person obtaining a
@@ -27,7 +27,7 @@
 * authorization.
*****/

```

```

/*
- * $Id: inchs.c,v 1.19 2022/12/11 00:01:39 tom Exp $
+ * $Id: inchs.c,v 1.20 2024/10/06 21:17:54 tom Exp $
*
* Author: Thomas E Dickey
*/
@@ -71,7 +71,7 @@
static int
test_inchs(int level, char **argv, WINDOW *chrwin, WINDOW *strwin)
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
    "Test input from screen using inch(), etc., in a moveable viewport.",
    ""
}
Index: test/ins_wide.c
Prereq: 1.30
--- ncurses-6.5-20240928+/test/ins_wide.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/ins_wide.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey
+ * Copyright
2020-2022,2024 Thomas E. Dickey
* Copyright 2002-2016,2017 Free Software Foundation, Inc.
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -27,7 +27,7 @@
* authorization.
*****/
/*
- * $Id: ins_wide.c,v 1.30 2022/12/10 22:28:50 tom Exp $
+ * $Id: ins_wide.c,v 1.31 2024/10/05 19:26:24 tom Exp $
*
* Demonstrate the wins_wstr() and wins_wch functions.
* Thomas Dickey - 2002/11/23
@@ -73,7 +73,7 @@
static int n_opt = -1;

static void
-legend(WINDOW *win, int level, Options state, wchar_t *buffer, int length)
+legend(WINDOW *win, int level, Options state, const wchar_t *buffer, int length)
{
    const char *showstate;

Index: test/insdelln.c
Prereq: 1.15
--- ncurses-6.5-20240928+/test/insdelln.c 2022-12-10

```



```

23:31:31.000000000 +0000
+++ ncurses-6.5-20241006/test/insdelln.c 2024-10-06 21:08:52.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 1919-2020,2022 Thomas E. Dickey                      *
+ * Copyright 1919-2022,2024 Thomas E. Dickey                      *
* Copyright 2008-2014,2017 Free Software Foundation, Inc.          *
*                                                                    *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.                                                    *
*****/
/*
- * $Id: insdelln.c,v 1.15 2022/12/10 23:31:31 tom Exp $
+ * $Id: insdelln.c,v 1.17 2024/10/06 21:08:52 tom Exp $
*
* test-driver for deleteln, wdeleteln, insdelln, winsdelln, insertln, winsertln
*/
@@ -160,7 +160,7 @@
}

static void
-do_subwindow(WINDOW *win, STATUS *sp, void func(WINDOW *))
+do_subwindow(WINDOW *win, const STATUS *sp, void func(WINDOW *))
{
    WINDOW *win1 = newwin(sp->y_max - 2, sp->x_max - 2,
        sp->y_beg + 1, sp->x_beg + 1);
@@ -204,7 +204,7 @@
static void
show_help(WINDOW *win)
{
-    static const char *table[] =
+    static NCURSES_CONST char *table[] =
    {
        "Basic commands:"
        ,"Use h/j/k/l or arrow keys to move the cursor."
Index: test/inserts.c
Prereq: 1.34
--- ncurses-6.5-20240928+/test/inserts.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/inserts.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey                      *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                      *
* Copyright 2002-2016,2017 Free Software Foundation, Inc.          *
*                                                                    *
*                                                                    *
* Permission is hereby granted, free of charge, to any person obtaining a *

```

```

@@ -27,7 +27,7 @@
* authorization.
*****/
/*
- * $Id: inserts.c,v 1.34 2022/12/10 22:28:50 tom Exp $
+ * $Id: inserts.c,v 1.35 2024/10/05 19:26:24 tom Exp $
*
* Demonstrate the winsstr() and winsch functions.
* Thomas Dickey - 2002/10/19
@@ -98,7 +98,7 @@
}

static int
-ColOf(char *buffer, int length, int margin)
+ColOf(const char *buffer, int length, int margin)
{
    int n;
    int result;
Index: test/knight.c
Prereq: 1.52
--- ncurses-6.5-20240928+/test/knight.c 2022-12-04 00:40:11.000000000 +0000
+++ ncurses-6.5-20241006/test/knight.c 2024-10-05 19:38:58.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright
2018-2021,2022 Thomas E. Dickey
+ * Copyright 2018-2022,2024 Thomas E. Dickey
* Copyright 1998-2013,2017 Free Software Foundation, Inc.
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -34,7 +34,7 @@
* Eric S. Raymond <esr@snark.thysus.com> July 22 1995. Mouse support
* added September 20th 1995.
*
- * $Id: knight.c,v 1.52 2022/12/04 00:40:11 tom Exp $
+ * $Id: knight.c,v 1.53 2024/10/05 19:38:58 tom Exp $
*/

#include <test.priv.h>
@@ -450,7 +450,8 @@
}

static bool
-evaluate_move(SQUARES squares, HISTORY * doneData, int doneSize, int row, int column)
+evaluate_move(SQUARES squares, const HISTORY * doneData, int doneSize,
+    int row, int column)
{
    if (doneSize <= 1)

```

```

return (TRUE);
Index: test/list_keys.c
Prereq: 1.33
--- ncurses-6.5-20240928+/test/list_keys.c 2023-11-11
00:35:05.000000000 +0000
+++ ncurses-6.5-20241006/test/list_keys.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2018-2022,2023 Thomas E. Dickey                *
+ * Copyright 2018-2023,2024 Thomas E. Dickey                *
* Copyright 2016,2017 Free Software Foundation, Inc.        *
*                                                              *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@@ -27,7 +27,7 @@@
* authorization.                                           *
*****/
/*
- * $Id: list_keys.c,v 1.33 2023/11/11 00:35:05 tom Exp $
+ * $Id: list_keys.c,v 1.34 2024/10/05 19:26:24 tom Exp $
*
* Author: Thomas E Dickey
*
@@@ -297,7 +297,7 @@@
#if NCURSES_XNAMES
    if (x_opt) {
        for (k = 0; k < count; ++k) {
-         TERMTYPE *term;
+         const TERMTYPE *term;
            set_curterm(terms[k]);
            term = (TERMTYPE *) cur_term;
            total += (size_t) (NUM_STRINGS(term) - STRCOUNT);
@@@ -348,7 +348,7 @@@
        widths1 = (int) strlen(modifier);

        for (k = 0; k < count; ++k) {
- char *value;
+ const char *value;
            set_curterm(terms[k]);
            if ((value = termname()) == NULL)
                failed("termname");
Index: test/move_field.c
Prereq: 1.16
--- ncurses-6.5-20240928+/test/move_field.c 2024-06-29 17:45:02.000000000 +0000
+++ ncurses-6.5-20241006/test/move_field.c 2024-10-06 21:12:35.000000000 +0000
@@@ -26,7 +26,7 @@@
* authorization.                                           *
*****/
/*

```

```

- * $Id: move_field.c,v 1.16 2024/06/29 17:45:02 tom Exp $
+ * $Id: move_field.c,v 1.19 2024/10/06 21:12:35 tom Exp $
*
* Demonstrate move_field().
*/
@@ -146,27 +146,27 @@
}

static FieldAttrs
*
-my_field_attrs(FIELD *f)
+my_field_attrs(NCURSES_CONST FIELD *f)
{
    return (FieldAttrs *) field_userptr(f);
}

static int
-buffer_length(FIELD *f)
+buffer_length(NCURSES_CONST FIELD *f)
{
    return my_field_attrs(f)->row_lengths[0];
}

static void
-set_buffer_length(FIELD *f, int length)
+set_buffer_length(NCURSES_CONST FIELD *f, int length)
{
    my_field_attrs(f)->row_lengths[0] = length;
}

static int
-offset_in_field(FORM *form)
+offset_in_field(NCURSES_CONST FORM *form)
{
-    FIELD *field = current_field(form);
+    NCURSES_CONST FIELD *field = current_field(form);
    int currow, curcol;

    form_getyx(form, currow, curcol);
@@ -245,7 +245,7 @@
}

default:
    modified = (ch < MIN_FORM_COMMAND
-    && isprint(ch));
+    && isprint(UChar(ch)));
    break;
}

```

@@ -275,9 +275,9 @@

/* display a status message to show what's happening */

static void

-show_status(FORM *form, FIELD *field)

+show_status(NCURSES_CONST FORM *form, NCURSES_CONST FIELD
*field)

{

- WINDOW *sub = form_sub(form);

+ NCURSES_CONST WINDOW *sub = form_sub(form);

int currow, curcol;

getyx(stdscr, currow, curcol);

Index: test/movewindow.c

Prereq: 1.54

--- ncurses-6.5-20240928+/test/movewindow.c 2023-05-27 20:13:10.000000000 +0000

+++ ncurses-6.5-20241006/test/movewindow.c 2024-10-06 18:44:26.000000000 +0000

@@ -1,5 +1,5 @@

/******

- * Copyright 2018-2022,2023 Thomas E. Dickey *

+ * Copyright 2018-2023,2024 Thomas E. Dickey *

* Copyright 2006-2013,2017 Free Software Foundation, Inc. *

* *

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -27,7 +27,7 @@

* authorization. *

*****/

/*

- * \$Id: movewindow.c,v 1.54 2023/05/27 20:13:10 tom Exp \$

+ * \$Id: movewindow.c,v 1.56 2024/10/06 18:44:26 tom Exp \$

*

* Demonstrate move functions for windows and derived windows from the curses

* library.

@@ -135,7 +135,7 @@

*/

static PAIR *

selectcell(WINDOW *parent,

- WINDOW *child,

+ NCURSES_CONST WINDOW *child,

int uli, int ulj,

int lri, int lrj,

bool relative,

@@ -252,7 +252,7 @@

int max_col = (parent == stdscr) ? COL_MAX : getmaxx(parent);

int min_line = (parent == stdscr) ? LINE_MIN : 0;

int max_line = (parent == stdscr) ? LINE_MAX : getmaxy(parent);

- PAIR *tmp;

+ NCURSES_CONST PAIR *tmp;

```

bool result = FALSE;

    head_line("Use arrows to move cursor, anything else to mark corner 1");
@@ -346,7 +346,7 @@
}

static WINDOW *
-parent_of(WINDOW *win)
+parent_of(NCURSES_CONST WINDOW *win)
{
    WINDOW *result = 0;
    int n = window2num(win);
@@ -435,7 +435,7 @@
int max_col = top ? COL_MAX
: getmaxx(parent);
int min_line = top ? LINE_MIN : 0;
int max_line = top ? LINE_MAX : getmaxy(parent);
- PAIR *tmp;
+ NCURSES_CONST PAIR *tmp;
bool more;

    head_line("Select new position for %swindow", top ? "" : "sub");
@@ -474,7 +474,7 @@
}

static void
-show_derwin(WINDOW *win)
+show_derwin(NCURSES_CONST WINDOW *win)
{
    int pary, parx, maxy, maxx;

@@ -500,7 +500,7 @@
int max_col = top ? COL_MAX : getmaxx(parent);
int min_line = top ? LINE_MIN : 0;
int max_line = top ? LINE_MAX : getmaxy(parent);
- PAIR *tmp;
+ NCURSES_CONST PAIR *tmp;
bool more;

    show_derwin(win);
Index: test/ncurses.c
Prereq: 1.538
--- ncurses-6.5-20240928+/test/ncurses.c 2023-11-11 01:23:59.000000000 +0000
+++ ncurses-6.5-20241006/test/ncurses.c 2024-10-06 20:18:59.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2018-2022,2023 Thomas E. Dickey
+ * Copyright 2018-2023,2024

```

Thomas E. Dickey *

* Copyright 1998-2016,2017 Free Software Foundation, Inc. *

* *

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -41,7 +41,7 @@

Author: Eric S. Raymond <esr@snark.thyrsus.com> 1993

Thomas E. Dickey (beginning revision 1.27 in 1996).

-\$Id: ncurses.c,v 1.538 2023/11/11 01:23:59 tom Exp \$

+\$Id: ncurses.c,v 1.541 2024/10/06 20:18:59 tom Exp \$

*****/

@@ -273,7 +273,7 @@

```

++x;
break;
default:
- if (!isprint(ch) || ch >= KEY_MIN) {
+ if (!isprint(UChar(ch)) || ch >= KEY_MIN) {
beep();
} else if ((int) strlen(buffer) < limit) {
int j;
@@ -928,7 +928,7 @@
/* at least Solaris SVR4 curses breaks unctrl(128), etc. */
c2 &= 0x7f;
#endif
- if (isprint(c))
+ if (isprint(UChar(c)))
(void)
wprintw(win, "%c", UChar(c));
else if (c2 != UChar(c))
(void) wprintw(win, "M-%s", unctrl(c2));
@@ -1693,7 +1693,7 @@
chtype extras = (chtype) ac;

if (UseColors) {
- NCURSES_PAIRS_T pair = 0;
+ NCURSES_PAIRS_T pair;
if ((fg != COLOR_BLACK) || (bg != COLOR_BLACK)) {
pair = 1;
if (init_pair(pair, fg, bg) == ERR) {
@@ -2345,7 +2345,6 @@
for (i = (NCURSES_PAIRS_T) (base_row * per_row); i < pairs_max; i++) {
int row = grid_top + (i / per_row) - base_row;
int col = (i % per_row + 1) * width;
- NCURSES_PAIRS_T pair = i;

if ((i / per_row) > row_limit)

```

```

break;
@@ -2355,6 +2354,7 @@
    if (row >= 0 && move(row, col) != ERR) {
        NCURSES_COLOR_T fg = (NCURSES_COLOR_T) InxToFG(i);
        NCURSES_COLOR_T bg = (NCURSES_COLOR_T) InxToBG(i);
+   NCURSES_PAIRS_T pair = i;

        init_pair(pair, fg, bg);
        attron(COLOR_PAIR(pair));
@@ -2517,7 +2517,7 @@
static int
x_color_test(bool recur GCC_UNUSED)
{
-   long i;
+   int i;

    int top = 0, width;
    int base_row = 0;
    int grid_top = top + 3;
@@ -2613,17 +2613,17 @@
/* show color names/numbers across the top */
for (i = 0; i < per_row; i++) {
    show_color_name(top + 2,
-   ((int) i + 1) * width,
-   (int) i * zoom_size + MinColors,
+   (i + 1) * width,
+   i * zoom_size + MinColors,
    opt_wide,
    opt_zoom);
}

/* show a grid of colors, with color names/ numbers on the left */
for (i = (base_row * per_row); i < pairs_max; i++) {
-   int row = grid_top + ((int) i / per_row) - base_row;
-   int col = ((int) i % per_row + 1) * width;
-   int pair = (int) i;
+   int row = grid_top + (i / per_row) - base_row;
+   int col = (i % per_row + 1) * width;
+   int pair = i;

    if ((i / per_row) > row_limit)
        break;
@@ -2828,7 +2828,7 @@
@@ -2828,7 +2828,7 @@
#define DecodeRGB(n) (NCURSES_COLOR_T) ((n * 1000) / 0xffff)

static void
-init_all_colors(bool xterm_colors, char *palette_file)
+init_all_colors(bool xterm_colors, NCURSES_CONST

```



```

char *palette_file)
{
    NCURSES_PAIRS_T cp;
    all_colors = typeMalloc(RGB_DATA, (unsigned) MaxColors);
@@ -3317,7 +3317,7 @@
{
    int c, fmt = 1;
    char buf[9];
-   char *s;
+   NCURSES_CONST char *s;
    attr_t attr = A_NORMAL;
    unsigned at_code = 0;
#if HAVE_SLK_COLOR
@@ -3938,8 +3938,7 @@

    *dst = *src;
    do {
-   int count;
-   TEST_CCHAR(src, count, {
+   TEST_CCHAR(src, {
        attr |= (test_attrs & A_ALTCHARSET);
        setcchar(dst, test_wch, attr, pair, NULL);
    }, {
@@ -4810,7 +4809,8 @@
/* Ask user for a window definition */
{
    WINDOW *rwindow;
-   pair ul, lr, *tmp;
+   pair ul, lr;
+   NCURSES_CONST pair *tmp;

    move(0, 0);
    clrtoeol();
@@ -5014,7 +5014,8 @@
#if HAVE_WRESIZE
    case CTRL('X'): /* resize window */
        if (current) {
-       pair *tmp, ul, lr;
+       NCURSES_CONST pair *tmp;
+       pair ul, lr;
            int mx, my;

            move(0, 0);
@@ -5249,7 +5250,7 @@
}

static void
-fill_panel(PANEL *pan)

```

```

+fill_panel(NCURSES_CONST PANEL
*pan)
{
    WINDOW *win = panel_window(pan);
    const char *userptr = (const char *) panel_userptr(pan);
@@ -5286,7 +5287,7 @@
}

static void
-fill_wide_panel(PANEL *pan)
+fill_wide_panel(NCURSES_CONST PANEL *pan)
{
    WINDOW *win = panel_window(pan);
    const char *userptr = (const char *) panel_userptr(pan);
@@ -5336,7 +5337,7 @@
}

static int
-demo_panels(void (*InitPanel) (WINDOW *), void (*FillPanel) (PANEL *))
+demo_panels(void (*InitPanel) (WINDOW *), void (*FillPanel) (NCURSES_CONST PANEL *))
{
    int count;
    int itmp;
@@ -6246,7 +6247,8 @@
run_trace_menu(MENU * m)
{
    ITEM **items;
-   ITEM *i, **p;
+   NCURSES_CONST ITEM *i;
+   ITEM **p;

    for (;;) {
        bool changed = FALSE;
@@ -6424,7 +6426,7 @@

        if (field_info(me, &rows, &cols, &frow, &fcol, &nrow, &nbuf) == E_OK
&& nbuf > 0) {
-   char *source = field_buffer(me, 1);
+   NCURSES_CONST char *source = field_buffer(me, 1);
        size_t have = (source ? strlen(source) : 0) + 1;
        size_t need
        = 80 + have;
        char *temp = malloc(need);
@@ -6434,7 +6436,7 @@
        _nc_STRNCPY(temp, source ? source : "", have + 1);
        len = (size_t) (char *) field_userptr(me);
        if (c <= KEY_MAX) {
-   if (isgraph(c) && (len + 1) < sizeof(temp)) {

```

```

+ if (isgraph(UChar(c)) && (len + 1) < sizeof(temp)) {
    temp[len++] = (char) c;
    temp[len] = 0;
    set_field_buffer(me, 1, temp);
@@ -6484,7 +6486,7 @@
}

static int
-form_virtualize(FORM *f, WINDOW *w)
+form_virtualize(NCURSES_CONST FORM *f, WINDOW *w)
{
    /* *INDENT-OFF* */
    static const struct {
@@ -6658,7 +6660,7 @@
static
FIELDCHECK_CB(pw_field_check)
{
- char *s = field_buffer(fld, 0);
+ NCURSES_CONST char *s = field_buffer(fld, 0);
    int n;

    for (n = 0; s[n] != '\0'; ++n) {
@@ -6680,7 +6682,8 @@
form_test(bool recur GCC_UNUSED)
{
    FORM *form;
- FIELD *f[12], *secure;
+ FIELD *f[12];
+ NCURSES_CONST FIELD *secure;
    FIELDTYPE *fty_middle = new_fieldtype(mi_field_check, mi_char_check);
    FIELDTYPE *fty_passwd = new_fieldtype(pw_field_check, pw_char_check);
    int c;
@@ -7176,7 +7179,7 @@
}

static void
-overlap_test_4(int flavor, WINDOW *a, WINDOW *b)
+overlap_test_4(int flavor, NCURSES_CONST WINDOW *a, WINDOW *b)
{
    switch ((otCOPY) flavor) {
        case otCOPY_overwrite:
@@ -7899,7 +7902,7 @@
        bool monochrome = FALSE;
#if HAVE_COLOR_CONTENT
        bool xterm_colors = FALSE;
- char *palette_file = 0;
+ NCURSES_CONST char *palette_file = 0;
#endif

```

```

    setlocale(LC_ALL, "");
Index: test/padview.c
Prereq: 1.22
--- ncurses-6.5-20240928+/test/padview.c 2022-12-04 00:40:11.000000000 +0000
+++ ncurses-6.5-20241006/test/padview.c 2024-10-06 21:08:25.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2019-2021,2022 Thomas E. Dickey                *
+ * Copyright 2019-2022,2024 Thomas E. Dickey                *
 * Copyright 2017 Free
Software Foundation, Inc.                                *
*                                                         *
 * Permission is hereby granted, free of charge, to any person obtaining a *
@@ -29,7 +29,7 @@
/*
 * clone of view.c, using pads
 *
- * $Id: padview.c,v 1.22 2022/12/04 00:40:11 tom Exp $
+ * $Id: padview.c,v 1.24 2024/10/06 21:08:25 tom Exp $
 */

#include <test.priv.h>
@@ -182,7 +182,7 @@

#if USE_WIDEC_SUPPORT
    if (!memcmp("\357\273\277", my_blob, 3)) {
- char *s = my_blob + 3;
+ const char *s = my_blob + 3;
    char *d = my_blob;
    Trace(("trim BOM"));
    do {
@@ -213,7 +213,7 @@
        for (k = 0; my_vec[k]; ++k) {
            char *s;
#if USE_WIDEC_SUPPORT
- char *last = my_vec[k] + (int) strlen(my_vec[k]);
+ const char *last = my_vec[k] + (int) strlen(my_vec[k]);
            wchar_t wch[2];
            size_t rc;
#ifdef state_unused
@@ -279,7 +279,7 @@
@@ -279,7 +279,7 @@
int
main(int argc, char *argv[])
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =

```

```

{
"Commands:",
" q,^Q,ESC - quit this program",
@@ -532,7 +532,7 @@
    beep();
    break;
}
- if (c >= KEY_MIN || (c > 0 && !isdigit(c))) {
+ if (c >= KEY_MIN || (c > 0 && !isdigit(UChar(c)))) {
    got_number = FALSE;
    value = 0;
}
Index: test/parse_rgb.h
Prereq: 1.5
--- ncurses-6.5-20240928+/test/parse_rgb.h 2020-02-02 23:34:34.000000000 +0000
+++ ncurses-6.5-20241006/test/parse_rgb.h 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2018,2020 Thomas E. Dickey
+ * Copyright 2018-2020,2024 Thomas E. Dickey
*
* Permission is hereby granted, free of charge, to any person obtaining a
* copy of this software and associated documentation files (the
@@ -26,7 +26,7 @@
* authorization.
*
*****/
/*
- * $Id: parse_rgb.h,v 1.5 2020/02/02 23:34:34 tom Exp $
+ * $Id: parse_rgb.h,v 1.6 2024/10/05 19:26:24 tom Exp $
*
* Sample implementation of ncurses RGB extension from user_caps(5).
*/
@@ -49,7 +49,7 @@
int bits;
int pwr2;
int r = 0, g = 0, b = 0;
- char *data;
+ const char *data;
char ch;

for (max_bits = 0, pwr2 = 1;
Index: test/picsmap.c
Prereq: 1.149
--- ncurses-6.5-20240928+/test/picsmap.c 2023-04-23 23:20:37.000000000 +0000
+++ ncurses-6.5-20241006/test/picsmap.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****

```

```

- * Copyright 2018-2022,2023 Thomas E. Dickey                *
+ * Copyright 2018-2023,2024 Thomas E. Dickey                *
* Copyright 2017,2018 Free Software Foundation, Inc.          *
*
*
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: picsmap.c,v 1.149 2023/04/23 23:20:37 tom Exp $
+ * $Id: picsmap.c,v 1.150 2024/10/05 19:26:24 tom Exp $
*
* Author: Thomas E. Dickey
*
@@ -574,7 +574,7 @@
    _nc_STRCAT(full_name, filename, need);
    if (tries & 4) {
        char *t = s;
- char *tc;
+ const char *tc;
        int num;
        char chr;
        int found = 0;
@@ -1374,8 +1374,8 @@
        int r, g, b, nocolor;
        float rf, gf, bf;
        unsigned check;
- char *t;
- char *s = t = strchr(buffer, '#');
+ char *s = strchr(buffer, '#');
+ const char *t = s;
        bool matched = FALSE;

        if (s != 0) {
@@ -1492,7 +1492,7 @@
#define fg_color(pics,n) (pics->fgcol[n].fgcol)

static void
-dump_picture(PICS_HEAD
+dump_picture(const PICS_HEAD * pics)
* pics)
{
    int y, x;

@@ -1767,7 +1767,7 @@
#if USE_EXTENDED_COLORS
case 'x':

```

```

    {
- char *s = optarg;
+ const char *s = optarg;
    while (*s) {
        switch (*s++) {
            case 'p':
Index: test/popup_msg.c
Prereq: 1.12
--- ncurses-6.5-20240928+/test/popup_msg.c 2024-06-29 18:34:58.000000000 +0000
+++ ncurses-6.5-20241006/test/popup_msg.c 2024-10-06 21:07:01.000000000 +0000
@@ -27,7 +27,7 @@
 * authorization.
 *
*****
/*
- * $Id: popup_msg.c,v 1.12 2024/06/29 18:34:58 tom Exp $
+ * $Id: popup_msg.c,v 1.13 2024/10/06 21:07:01 tom Exp $
 *
 * Show a multi-line message in a window which may extend beyond the screen.
 *
@@ -62,7 +62,7 @@
 * Display a temporary window, e.g., to display a help-message.
 */
void
-popup_msg(WINDOW *parent, const char *const *msg)
+popup_msg(WINDOW *parent, NCURSES_CONST
char *const *msg)
{
    int x0 = 4;
    int y0 = 2;
@@ -171,12 +171,12 @@
void
popup_msg2(WINDOW *parent, char **msg)
{
-    popup_msg(parent, (const char *const *) msg);
+    popup_msg(parent, (NCURSES_CONST char *const *) msg);
}

#else
void
-popup_msg(WINDOW *parent, const char *const *msg)
+popup_msg(WINDOW *parent, NCURSES_CONST char *const *msg)
{
    (void) parent;
    (void) msg;
Index: test/popup_msg.h
Prereq: 1.5
--- ncurses-6.5-20240928+/test/popup_msg.h 2020-02-02 23:34:34.000000000 +0000
+++ ncurses-6.5-20241006/test/popup_msg.h 2024-10-06 21:17:54.000000000 +0000

```

```

@@ -1,5 +1,5 @@
/*****

- * Copyright 2018,2020 Thomas E. Dickey                *
+ * Copyright 2018-2020,2024 Thomas E. Dickey            *
* Copyright 2017 Free Software Foundation, Inc.          *
*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: popup_msg.h,v 1.5 2020/02/02 23:34:34 tom Exp $
+ * $Id: popup_msg.h,v 1.6 2024/10/06 21:17:54 tom Exp $
*
* Utility functions for a popup-message or help-screen.
*/
@@ -37,7 +37,7 @@

#include <test.priv.h>

-extern void popup_msg(WINDOW *parent, const char *const *msg);
+extern void popup_msg(WINDOW *parent, NCURSES_CONST char *const *msg);
extern void popup_msg2(WINDOW *parent, char **msg);

#ifdef /* POPUP_MSG_H_incl */
Index: test/railroad.c
Prereq: 1.26
--- ncurses-6.5-20240928+/test/railroad.c 2022-12-11 00:12:13.000000000 +0000
+++ ncurses-6.5-20241006/test/railroad.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****

-
* Copyright 2019-2020,2022 Thomas E. Dickey                *
+ * Copyright 2019-2022,2024 Thomas E. Dickey            *
* Copyright 2000-2013,2017 Free Software Foundation, Inc.    *
*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
@@
/*
* Author: Thomas E. Dickey - 2000
*
- * $Id: railroad.c,v 1.26 2022/12/11 00:12:13 tom Exp $
+ * $Id: railroad.c,v 1.27 2024/10/05 19:26:24 tom Exp $
*
* A simple demo of the termcap interface.
*/
@@ -113,7 +113,7 @@

```



```

static void
ShowSign(char *string)
{
-   char *base = string;
+   const char *base = string;
    int first, last;

    if (moveit != 0) {
Index: test/rain.c
Prereq: 1.58
--- ncurses-6.5-20240928+/test/rain.c 2024-08-31 15:56:10.000000000 +0000
+++ ncurses-6.5-20241006/test/rain.c 2024-10-06 21:09:00.000000000 +0000
@@ -27,7 +27,7
@@
* authorization.
*****/
/*
- * $Id: rain.c,v 1.58 2024/08/31 15:56:10 tom Exp $
+ * $Id: rain.c,v 1.62 2024/10/06 21:09:00 tom Exp $
*/
#include <test.priv.h>
#include <popup_msg.h>
@@ -45,7 +45,7 @@

struct DATA;

-typedef void (*DrawPart) (struct DATA *);
+typedef void (*DrawPart) (const struct DATA *);

typedef struct DATA {
    int y, x;
@@ -118,25 +118,25 @@
}

static void
-part1(DATA * drop)
+part1(const DATA * drop)
{
    MvAddCh(drop->y, drop->x, '.');
}

static void
-part2(DATA * drop)
+part2(const DATA * drop)
{
    MvAddCh(drop->y, drop->x, 'o');
}

```

```

static void
-part3(DATA * drop)
+part3(const DATA * drop)
{
    MvAddCh(drop->y, drop->x, 'O');
}

```

```

static void
-part4(DATA * drop)
+part4(const DATA * drop)
{
    MvAddCh(drop->y - 1, drop->x, '-');
    MvAddStr(drop->y, drop->x - 1, "|.");
@@ -144,7 +144,7 @@
}

```

```

static void
-part5(DATA * drop)
+part5(const DATA * drop)
{
    MvAddCh(drop->y - 2, drop->x, '-');
    MvAddStr(drop->y - 1, drop->x - 1, "/\\");
@@ -154,7 +154,7 @@
}

```

```

static void
-part6(DATA * drop)
+part6(const DATA * drop)
{
    MvAddCh(drop->y - 2, drop->x, ' ');
    MvAddStr(drop->y - 1, drop->x - 1, " ");
@@ -186,7 +186,7 @@
}

```

```

static void
-draw_part(void (*func) (DATA *), int state, DATA * data)
+draw_part(void (*func) (const DATA *), int state, DATA * data)
{
    data->func = func;
    data->state = state;
@@ -328,7 +328,7 @@
int
main(int argc, char *argv[])
{
-    static const char *help[] =
+    static NCURSES_CONST char *help[] =
    {
        "Commands:",

```

```

" q/Q      exit the program",
Index: test/redraw.c
Prereq: 1.17
--- ncurses-6.5-20240928+/test/redraw.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/redraw.c 2024-10-06 21:17:54.000000000 +0000
@@@ -1,5 +1,5 @@@
/*****
-
* Copyright 2020-2021,2022 Thomas E. Dickey                *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                *
* Copyright 2006-2012,2017 Free Software Foundation, Inc.    *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@@ -27,7 +27,7 @@@
* authorization.
*****/
/*
- * $Id: redraw.c,v 1.17 2022/12/10 22:28:50 tom Exp $
+ * $Id: redraw.c,v 1.18 2024/10/06 21:17:54 tom Exp $
*
* Demonstrate the redrawwin() and wredrawln() functions.
* Thomas Dickey - 2006/11/4
@@@ -61,7 +61,7 @@@
static void
test_redraw(WINDOW *win)
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
"Commands:",
" ^Q/ESC/q - quit",
Index: test/savescreen.c
Prereq:
1.62
--- ncurses-6.5-20240928+/test/savescreen.c 2022-12-10 23:23:27.000000000 +0000
+++ ncurses-6.5-20241006/test/savescreen.c 2024-10-06 21:10:09.000000000 +0000
@@@ -1,5 +1,5 @@@
/*****
- * Copyright 2018-2021,2022 Thomas E. Dickey                *
+ * Copyright 2018-2022,2024 Thomas E. Dickey                *
* Copyright 2006-2017,2018 Free Software Foundation, Inc.    *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@@ -27,7 +27,7 @@@
* authorization.
*****/
/*
- * $Id: savescreen.c,v 1.62 2022/12/10 23:23:27 tom Exp $

```

```

+ * $Id: savescreen.c,v 1.65 2024/10/06 21:10:09 tom Exp $
*
* Demonstrate save/restore functions
from the curses library.
* Thomas Dickey - 2007/7/14
@@ -96,7 +96,7 @@
}

static int
-load_screen(char *filename)
+load_screen(NCURSES_CONST char *filename)
{
    int result;

@@ -170,7 +170,7 @@
#if USE_WIDEC_SUPPORT
    cchar_t mycc;
#endif
-   char *filename = files[which];
+   NCURSES_CONST char *filename = files[which];
    bool dumped = FALSE;

    if (filename != 0) {
@@ -216,7 +216,7 @@
static void
editor_help(void)
{
-   static const char *msgs[] =
+   static NCURSES_CONST char *msgs[] =
    {
        "You are now in the screen-editor, which allows you to make some",
        "lines on the screen, as well as save copies of the screen to a",
@@ -238,7 +238,7 @@
static void
replay_help(void)
{
-   static const char *msgs[] =
+   static NCURSES_CONST char *msgs[] =
    {
        "You are now in the screen-loader, which allows you to view",
        "the dumped/restored screens.",
@@ -286,7 +286,7 @@
    bool replaying = FALSE;
    bool done = FALSE;
    char **files;
-
    char *fill_by = 0;
+   NCURSES_CONST char *fill_by = 0;

```

```

#if USE_WIDEC_SUPPORT
    cchar_t mycc;
    static const wchar_t mywc[2] =
Index: test/tclock.c
Prereq: 1.48
--- ncurses-6.5-20240928+/test/tclock.c 2023-02-25 16:42:22.000000000 +0000
+++ ncurses-6.5-20241006/test/tclock.c 2024-10-05 18:47:56.000000000 +0000
@@ -1,4 +1,4 @@
-/* $Id: tclock.c,v 1.48 2023/02/25 16:42:22 tom Exp $ */
+/* $Id: tclock.c,v 1.49 2024/10/05 18:47:56 tom Exp $ */

#define NEED_TIME_H
#include <test.priv.h>
@@ -145,7 +145,7 @@
    int lastbeep = -1;
    bool odd = FALSE;
    time_t tim;
-   struct tm *t;
+   const struct tm *t;
    char szChar[20];
    char *text;
    short my_bg = COLOR_BLACK;
Index: test/test.priv.h
Prereq: 1.219
--- ncurses-6.5-20240928+/test/test.priv.h 2024-08-31 10:46:01.000000000 +0000
+++ ncurses-6.5-20241006/test/test.priv.h 2024-10-05 19:01:00.000000000 +0000
@@ -30,7 +30,7 @@
/*****
* Author: Thomas E. Dickey          1996-on          *
*****/
-/* $Id: test.priv.h,v 1.219 2024/08/31 10:46:01 Rafael.Kitover Exp $ */
+/* $Id: test.priv.h,v 1.220 2024/10/05 19:01:00 tom Exp $ */

#ifndef __TEST_PRIV_H
#define __TEST_PRIV_H 1
@@ -765,8 +765,8 @@

#define VERSION_COMMON() \
static char *version_common(char **argv) { \
-   char *base = argv[0]; \
-   char *part = strrchr(base, '/'); \
+   const char *base = argv[0]; \
+   const char *part = strrchr(base, '/'); \
    size_t need = strlen(base) + 80; \
    char *result = malloc(need); \
    if (result != NULL) { \
@@ -916,8 +916,8 @@
* Workaround in case getcchar() returns a positive value when the source

```

```

* string produces only a L'\0'.
*/
#define TEST_CCHAR(s, count, then_stmt, else_stmt) \
- if ((count = getcchar(s, NULL, NULL, NULL, NULL)) > 0) { \
+ #define TEST_CCHAR(s, then_stmt, else_stmt) \
+ if (getcchar(s, NULL, NULL, \
NULL, NULL) > 0) { \
    wchar_t test_wch[CCHARW_MAX + 2]; \
    attr_t test_attrs; \
    NCURSES_PAIRS_T test_pair; \
Index: test/test_add_wchstr.c
Prereq: 1.34
--- ncurses-6.5-20240928+/test/test_add_wchstr.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/test_add_wchstr.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020-2021,2022 Thomas E. Dickey                      *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                      *
* Copyright 2009-2016,2017 Free Software Foundation, Inc.        *
*                                                                    *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.                                                  *
*****/
/*
- * $Id: test_add_wchstr.c,v 1.34 2022/12/10 22:28:50 tom Exp $
+ * $Id: test_add_wchstr.c,v 1.35 2024/10/05 19:26:24 tom Exp $
*
* Demonstrate the waddwchstr() and wadd_wch functions.
* Thomas Dickey - 2009/9/12
@@ -158,7 +158,7 @@
}

```

```

static void
-legend(WINDOW *win, int level, Options state, wchar_t *buffer, int length)
+legend(WINDOW *win, int level, Options state, const wchar_t *buffer, int length)
{
    const char *showstate;

```

```

Index: test/test_addchstr.c
Prereq: 1.29
--- ncurses-6.5-20240928+/test/test_addchstr.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/test_addchstr.c 2024-10-05 19:26:24.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey                      *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                      *
* Copyright 2009-2016,2017 Free Software Foundation, Inc.        *

```

```

*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: test_addchstr.c,v 1.29 2022/12/10 22:28:50 tom Exp $
+ * $Id: test_addchstr.c,v 1.30 2024/10/05 19:26:24 tom Exp $
*
* Demonstrate the waddchstr() and waddch functions.
* Thomas Dickey - 2009/9/12
@@ -169,7 +169,7 @@
}

static int
-ColOf(char *buffer, int length, int margin)
+ColOf(const char *buffer, int length, int margin)
{
    int n;
    int result;
Index: test/test_addstr.c
Prereq: 1.20
--- ncurses-6.5-20240928+/test/test_addstr.c 2022-12-10 22:28:50.000000000 +0000
+++ ncurses-6.5-20241006/test/test_addstr.c 2024-10-05 19:27:33.000000000 +0000
@@ -1,5 +1,5 @@
/*****
-
* Copyright 2020,2022 Thomas E. Dickey
*
+ * Copyright 2020-2022,2024 Thomas E. Dickey
*
* Copyright 2009-2016,2017 Free Software Foundation, Inc.
*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: test_addstr.c,v 1.20 2022/12/10 22:28:50 tom Exp $
+ * $Id: test_addstr.c,v 1.21 2024/10/05 19:27:33 tom Exp $
*
* Demonstrate the waddstr() and waddch functions.
* Thomas Dickey - 2009/9/12
@@ -91,7 +91,7 @@
}

static int
-ColOf(char *buffer, int length, int margin)
+ColOf(const char *buffer, int length, int margin)
{

```

```

    int n;
    int result;
Index: test/test_addwstr.c
Prereq: 1.22
---
ncurses-6.5-20240928+/test/test_addwstr.c 2023-05-27 20:13:10.000000000 +0000
+++ ncurses-6.5-20241006/test/test_addwstr.c 2024-10-05 19:27:33.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020-2022,2023 Thomas E. Dickey                *
+ * Copyright 2020-2023,2024 Thomas E. Dickey                *
 * Copyright 2009-2016,2017 Free Software Foundation, Inc.    *
 *                                                            *
 * Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
 * authorization.                                           *
*****/
/*
- * $Id: test_addwstr.c,v 1.22 2023/05/27 20:13:10 tom Exp $
+ * $Id: test_addwstr.c,v 1.23 2024/10/05 19:27:33 tom Exp $
 *
 * Demonstrate the waddwstr() and wadd_wch
functions.
 * Thomas Dickey - 2009/9/12
@@ -82,7 +82,7 @@
static int n_opt = -1;

static void
-legend(WINDOW *win, int level, Options state, wchar_t *buffer, int length)
+legend(WINDOW *win, int level, Options state, const wchar_t *buffer, int length)
{
    const char *showstate;

Index: test/test_endwin.c
Prereq: 1.2
--- ncurses-6.5-20240928+/test/test_endwin.c 2023-11-10 15:17:19.000000000 +0000
+++ ncurses-6.5-20241006/test/test_endwin.c 2024-10-05 19:27:33.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2023 Thomas E. Dickey                *
+ * Copyright 2023,2024 Thomas E. Dickey                *
 *                                                            *
 * Permission is hereby granted, free of charge, to any person obtaining a *
 * copy of this software and associated documentation files (the      *
@@ -26,7 +26,7 @@
 * authorization.                                           *
*****/
/*

```



```

- * $Id: test_endwin.c,v 1.2 2023/11/10 15:17:19 tom Exp $
+ * $Id: test_endwin.c,v 1.3 2024/10/05 19:27:33 tom Exp $
*/
#include <test.priv.h>

@@ -67,7 +67,7 @@
    int rc_e1 = OK;
    int rc_e2 = OK;
    int rc_e3 = OK;
-   SCREEN *sp = NULL;
+   const SCREEN *sp = NULL;
    bool opt_e = FALSE;
    bool opt_i = FALSE;
    bool opt_n = FALSE;
Index: test/test_get_wstr.c
Prereq: 1.14
--- ncurses-6.5-20240928+/test/test_get_wstr.c 2022-12-10 23:59:13.000000000 +0000
+++ ncurses-6.5-20241006/test/test_get_wstr.c 2024-10-06 23:15:10.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey                                *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                          *
 * Copyright 2007-2011,2017
Free Software Foundation, Inc.          *
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
 * authorization.
*****/
/*
- * $Id: test_get_wstr.c,v 1.14 2022/12/10 23:59:13 tom Exp $
+ * $Id: test_get_wstr.c,v 1.16 2024/10/06 23:15:10 tom Exp $
 *
 * Author: Thomas E Dickey
 *
@@ -69,7 +69,7 @@
}

static int
-Remainder(WINDOW *txtwin)
+Remainder(const WINDOW *txtwin)
{
    int result = getmaxx(txtwin) - getcurx(txtwin);
    return (result > 0) ? result : 0;
@@ -138,7 +138,7 @@
static int
recursive_test(int level, char **argv, WINDOW *strwin)
{

```

```

- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
"Commands:",
" q,^Q,ESC - quit this program",
Index: test/test_getstr.c
Prereq:
1.16
--- ncurses-6.5-20240928+/test/test_getstr.c 2022-12-10 23:58:37.000000000 +0000
+++ ncurses-6.5-20241006/test/test_getstr.c 2024-10-06 22:36:46.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020,2022 Thomas E. Dickey                      *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                  *
* Copyright 2007-2012,2017 Free Software Foundation, Inc.      *
*                                                                *
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -27,7 +27,7 @@
* authorization.
*****/
/*
- * $Id: test_getstr.c,v 1.16 2022/12/10 23:58:37 tom Exp $
+ * $Id: test_getstr.c,v 1.19 2024/10/06 22:36:46 tom Exp $
*
* Author: Thomas E Dickey
*
@@ -76,7 +76,7 @@
}

static int
-Remainder(WINDOW *txtwin)
+Remainder(NCURSES_CONST WINDOW *txtwin)
{
int result = getmaxx(txtwin) - getcurx(txtwin);
return (result > 0) ? result : 0;
@@ -145,7 +145,7 @@
static int
recursive_test(int level, char **argv, WINDOW *strwin)
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
"Commands:",
" q,^Q,ESC - quit this program",
Index: test/test_instr.c
Prereq: 1.13
--- ncurses-6.5-20240928+/test/test_instr.c 2024-06-29 18:42:53.000000000 +0000
+++ ncurses-6.5-20241006/test/test_instr.c 2024-10-06 21:11:54.000000000 +0000

```

```

@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: test_instr.c,v 1.13 2024/06/29 18:42:53 tom Exp $
+ * $Id: test_instr.c,v 1.15 2024/10/06 21:11:54 tom Exp $
*
* Author: Thomas E Dickey
*
@@ -56,13 +56,13 @@
}

static
void
-show_1st(WINDOW *win, int line, char *buffer)
+show_1st(WINDOW *win, int line, NCURSES_CONST char *buffer)
{
    MvWAddStr(win, line, 5, buffer);
}

static void
-showmore(WINDOW *win, int line, char *buffer)
+showmore(WINDOW *win, int line, NCURSES_CONST char *buffer)
{
    wmove(win, line, 0);
    wclrtoeol(win);
@@ -72,7 +72,7 @@
static void
show_help(WINDOW *win)
{
- static const char *msgs[] =
+ static NCURSES_CONST char *msgs[] =
{
    "Show file contents and a viewport from the variants of winstr."
    ,"Use h/j/k/l or arrow keys to move the viewport."
Index: test/test_inwstr.c
Prereq: 1.9
--- ncurses-6.5-20240928+/test/test_inwstr.c 2024-06-29 18:40:02.000000000 +0000
+++ ncurses-6.5-20241006/test/test_inwstr.c 2024-10-06 23:15:38.000000000 +0000
@@ -27,7 +27,7 @@
* authorization.
*****/

/*
- * $Id: test_inwstr.c,v
1.9 2024/06/29 18:40:02 tom Exp $
+ * $Id: test_inwstr.c,v 1.11 2024/10/06 23:15:38 tom Exp $
*
* Author: Thomas E Dickey

```

```

*
@@ -59,13 +59,13 @@
}

static void
-show_1st(WINDOW *win, int line, wchar_t *buffer)
+show_1st(WINDOW *win, int line, const wchar_t *buffer)
{
    (void) mvwaddwstr(win, line, 5, buffer);
}

static void
-showmore(WINDOW *win, int line, wchar_t *buffer)
+showmore(WINDOW *win, int line, const wchar_t *buffer)
{
    wmove(win, line, 0);
    wclrtoeol(win);
@@ -75,7 +75,7 @@
static void
show_help(WINDOW *win)
{
-    static const char *msgs[] =
+    static NCURSES_CONST char *msgs[] =
    {
        "Show file contents and a viewport from the variants of winwstr."
        ,"Use h/j/k/l or arrow keys to move the viewport."
Index: test/test_mouse.c
Prereq: 1.32
--- ncurses-6.5-20240928+/test/test_mouse.c 2024-08-31 10:46:01.000000000 +0000
+++ ncurses-6.5-20241006/test/test_mouse.c 2024-10-05 18:47:56.000000000 +0000
@@ -22,7 +22,7 @@
* THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
*
*****/
/*
- * $Id: test_mouse.c,v 1.32 2024/08/31 10:46:01 Rafael.Kitover Exp $
+ * $Id: test_mouse.c,v 1.33 2024/10/05 18:47:56 tom Exp $
*
* Author: Leonid S Usov
*
@@ -38,7 +38,7 @@
static void
raw_loop(void)
{
-    char *xtermcap;
+    const char *xtermcap;

    printf("Entering raw mode. Ctrl-c to quit.\n");

```

Index: test/test_opaque.c

Prereq: 1.15

--- ncurses-6.5-20240928+/test/test_opaque.c 2022-12-11 00:03:10.000000000 +0000

+++ ncurses-6.5-20241006/test/test_opaque.c 2024-10-05 19:27:33.000000000 +0000

@@ -1,5 +1,5 @@

/*****

- * Copyright 2020-2021,2022 Thomas E. Dickey *
+ * Copyright 2020-2022,2024 Thomas E. Dickey *
* Copyright 2007-2008,2009 Free Software Foundation, Inc. *
*

*

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -27,7 +27,7 @@

* authorization. *

*****/

/*

- * \$Id: test_opaque.c,v 1.15 2022/12/11 00:03:10 tom Exp \$

+ * \$Id: test_opaque.c,v 1.16 2024/10/05 19:27:33 tom Exp \$

*

* Author: Thomas E Dickey

*

@@ -157,13 +157,13 @@

}

static int

-status_y(WINDOW *stswin, int cell)

+status_y(const WINDOW *stswin, int cell)

{

return (cell % getmaxy(stswin));

}

static int

-status_x(WINDOW *stswin, int cell)

+status_x(const WINDOW *stswin, int cell)

{

return (15 * (cell / getmaxy(stswin)));

}

Index: test/test_sgr.c

Prereq: 1.22

--- ncurses-6.5-20240928+/test/test_sgr.c 2023-05-27 20:13:10.000000000 +0000

+++ ncurses-6.5-20241006/test/test_sgr.c 2024-10-06 21:05:50.000000000 +0000

@@ -1,5 +1,5 @@

/*****

-

* Copyright 2019-2022,2023 Thomas E. Dickey *
+ * Copyright 2019-2023,2024 Thomas E. Dickey *
* Copyright 2015-2016,2017 Free Software Foundation, Inc. *

```

*
*
* Permission is hereby granted, free of charge, to any person obtaining a *
@@ -30,7 +30,7 @@
/*
* Author: Thomas E. Dickey
*
- * $Id: test_sgr.c,v 1.22 2023/05/27 20:13:10 tom Exp $
+ * $Id: test_sgr.c,v 1.25 2024/10/06 21:05:50 tom Exp $
*
* A simple demo of the sgr/sgr0 terminal capabilities.
*/
@@ -142,10 +142,10 @@
#define MASK_REV (1 << 2)

static void
-dumpit(unsigned bits, unsigned ignore, const char *sgr, const char *sgr0)
+dumpit(unsigned bits, unsigned ignore, NCURSES_CONST char *sgr, NCURSES_CONST char *sgr0)
{
- static const char sample[] = "abcdefghijklm";
- static
char params[] = "SURBDBIPA";
+ static NCURSES_CONST char sample[] = "abcdefghijklm";
+ static NCURSES_CONST char params[] = "SURBDBIPA";
    unsigned n;

    printf("%4u ", bits);
@@ -177,12 +177,11 @@
static void
brute_force(const char *name)
{
- unsigned count;
- char *my_sgr;
- char *my_sgr0;
- char *my_bold;
- char *my_revs;
- char *my_smso;
+ NCURSES_CONST char *my_sgr;
+ NCURSES_CONST char *my_sgr0;
+ NCURSES_CONST char *my_bold;
+ NCURSES_CONST char *my_revs;
+ NCURSES_CONST char *my_smso;
    char *my_name = strdup(name);

    if (db_list) {
@@ -219,6 +218,7 @@
    unsigned ignore = 0;
    unsigned reason = 0;
    unsigned repeat = 0;

```

```

+ unsigned count;
for (count = 0; count < MAXSGR; ++count) {
    values[count] = tparm(my_sgr,
        BITS2P(1),
@@ -334,7 +334,7 @@
main(int argc, char *argv[])
{
    int ch;
-   char *name;
+   const char *name;

    while ((ch = getopt(argc, argv, OPTS_COMMON "d:e:nq")) != -1) {
switch (ch)
{
@@ -369,7 +369,7 @@
    } else if ((name = getenv("TERM")) != 0) {
brute_force(name);
    } else {
- static char dumb[] = "dumb";
+ static const char dumb[] = "dumb";
brute_force(dumb);
    }
}

```

Index: test/test_tparm.c

Prereq: 1.39

--- ncurses-6.5-20240928+/test/test_tparm.c 2023-11-11 01:00:03.000000000 +0000

+++ ncurses-6.5-20241006/test/test_tparm.c 2024-10-06 20:50:27.000000000 +0000

@@ -1,5 +1,5 @@

/*

*/

- * Copyright 2020-2022,2023 Thomas E. Dickey *

+ * Copyright 2020-2023,2024 Thomas E. Dickey *

*

*

* Permission is hereby granted, free of charge, to any person obtaining a *

* copy of this software and associated documentation files (the *

@@ -29,7 +29,7 @@

/*

* Author: Thomas E. Dickey

*

- * \$Id: test_tparm.c,v 1.39 2023/11/11 01:00:03

tom Exp \$

+ * \$Id: test_tparm.c,v 1.42 2024/10/06 20:50:27 tom Exp \$

*

* Exercise tparm/tparm, either for all possible capabilities with fixed

* parameters, or one capability with specific combinations of parameters.

@@ -107,7 +107,7 @@

}

static int

```

-isNumeric(char *source)
+isNumeric(NCURSES_CONST char *source)
{
    char *next = 0;
    long value = strtol(source, &next, 0);
@@ -245,7 +245,7 @@
#define NS_9(fmt) NS_8(fmt), NumStr(8)

static void
-test_tparm(const char *name, const char *format, long *number, char **string)
+test_tparm(const char *name, NCURSES_CONST char *format, long *number, char **string)
{
    char *use_strings[MAX_PARM];
    char *result = NULL;
@@ -527,7 +527,7 @@
    }
} else {
    char *s = t_opt;
-   char *t;
+   NCURSES_CONST char *t;
    while ((t = strtok(s, ",")) != 0) {
        s = 0;
        if (len_terms + 2 >= max_terms) {
@@ -603,7 +603,7 @@
#if NCURSES_XNAMES
    len_caps = std_caps;
    if (cur_term) {
-   TERMTYPE *term =
(TERMTYPE *) cur_term;
+   NCURSES_CONST TERMTYPE *term = (TERMTYPE *) cur_term;
    for (n = STRCOUNT; n < NUM_STRINGS(term); ++n) {
        GrowArray(all_caps, max_caps, len_caps);
        GrowArray(cap_name, max_name, len_caps);
Index: test/test_vidputs.c
Prereq: 1.15
--- ncurses-6.5-20240928+/test/test_vidputs.c 2022-12-10 23:23:27.000000000 +0000
+++ ncurses-6.5-20241006/test/test_vidputs.c 2024-10-06 21:17:54.000000000 +0000
@@ -1,5 +1,5 @@
/*****
- * Copyright 2020-2021,2022 Thomas E. Dickey                *
+ * Copyright 2020-2022,2024 Thomas E. Dickey                *
* Copyright 2013-2014,2017 Free Software Foundation, Inc.    *
*
* Permission is hereby granted, free of charge, to any person obtaining a
@@ -27,7 +27,7 @@
* authorization.
*
*****/

```



```

/*
- * $Id: test_vidputs.c,v 1.15 2022/12/10 23:23:27 tom Exp $
+ * $Id: test_vidputs.c,v 1.16 2024/10/06 21:17:54 tom Exp $
*
* Demonstrate the vidputs and vidattr functions.
* Thomas Dickey - 2013/01/12
@@ -51,7 +51,7 @@
}

static bool
-outs(const char *s)
+outs(NCURSES_CONST char *s)
{
    if (VALID_STRING(s)) {
        tputs(s, 1, outc);
Index: test/testcurs.c
Prereq: 1.58
--- ncurses-6.5-20240928+/test/testcurs.c 2023-05-28 14:23:34.000000000 +0000
+++ ncurses-6.5-20241006/test/testcurs.c 2024-10-05 19:17:19.000000000 +0000
@@ -6,7 +6,7 @@
* wrs(5/28/93) -- modified to be consistent (perform identically) with either
*
*      PDCurses or under Unix System V, R4
*
- * $Id: testcurs.c,v 1.58 2023/05/28 14:23:34 tom Exp $
+ * $Id: testcurs.c,v 1.59 2024/10/05 19:17:19 tom Exp $
*/

#include <test.priv.h>
@@ -245,7 +245,7 @@
    wclrtoobot(win);
    if (c >= KEY_MIN)
        wprintw(win, "Key Pressed: %s", keyname(c));
- else if (isprint(c))
+ else if (isprint(UChar(c)))
        wprintw(win, "Key Pressed: %c", c);
    else
        wprintw(win, "Key Pressed: %s", unctrl(UChar(c)));
@@ -748,8 +748,8 @@
    keypad(stdscr, TRUE);
    raw();
    key = getch();
- if (key < KEY_MIN && key > 0 && isalpha(key)) {
-     if (islower(key))
+ if (key < KEY_MIN && key > 0 && isalpha(UChar(key))) {
+     if (islower(UChar(key)))
        key = toupper(key);
        for (n = 0; n < MAX_OPTIONS; ++n) {
            if (key == command[n].text[0]) {

```

Index: test/testscanw.c

Prereq: 1.15

--- ncurses-6.5-20240928+/test/testscanw.c 2022-12-11 00:10:29.000000000 +0000

+++ ncurses-6.5-20241006/test/testscanw.c 2024-10-05 19:27:33.000000000 +0000

@@ -1,5 +1,5 @@

/*****

- * Copyright 1919-2020,2022 Thomas E. Dickey *

+ * Copyright 1919-2022,2024 Thomas E. Dickey *

*

Copyright 1998-2002,2006 Free Software Foundation, Inc. *

*

*

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -30,7 +30,7 @@

* Date: 1997/03/17

* From: bayern@morpheus.cis.yale.edu

*

- * \$Id: testscanw.c,v 1.15 2022/12/11 00:10:29 tom Exp \$

+ * \$Id: testscanw.c,v 1.16 2024/10/05 19:27:33 tom Exp \$

*/

#include <test.priv.h>

@@ -85,7 +85,7 @@

 curses_trace TRACE_UPDATE | TRACE_CALLS);

#endif

 while (optind < argc) {

- char *token = argv[optind++];

+ const char *token = argv[optind++];

 if (isdigit(UChar(*token)))

 move(atoi(token), 0);

 else if (!strcmp(token, "k+"))

Index: test/view.c

Prereq: 1.145

--- ncurses-6.5-20240928+/test/view.c 2022-12-04 00:40:11.000000000 +0000

+++ ncurses-6.5-20241006/test/view.c 2024-10-06 21:08:05.000000000 +0000

@@ -1,5 +1,5 @@

/*****

-

* Copyright 1919-2021,2022 Thomas E. Dickey *

+ * Copyright 1919-2022,2024 Thomas E. Dickey *

* Copyright 1998-2016,2017 Free Software Foundation, Inc. *

*

*

* Permission is hereby granted, free of charge, to any person obtaining a *

@@ -52,7 +52,7 @@

* scroll operation worked, and the refresh() code only had to do a

* partial repaint.

*

- * \$Id: view.c,v 1.145 2022/12/04 00:40:11 tom Exp \$

+ * \$Id: view.c,v 1.147 2024/10/06 21:08:05 tom Exp \$

```

*/

#include <test.priv.h>
@@ -93,8 +93,7 @@

#if USE_WIDEC_SUPPORT
    for (;;) {
- int count;
- TEST_CCHAR(src, count, {
+ TEST_CCHAR(src, {
    int len = wcwidth(test_wch[0]);
    result += (len > 0) ? len : 1;
    ++src;
@@ -186,12 +185,10 @@
    */
    {
    int j;
- int width = 1;
+ int width;

    for (j = actual = 0; j < shift; ++j) {
- int count;
-
-
    TEST_CCHAR(s + j, count, {
+ TEST_CCHAR(s + j, {
    width = wcwidth(test_wch[0]);
    }
    , {
@@ -294,7 +291,7 @@

#if USE_WIDEC_SUPPORT
    if (!memcmp("\357\273\277", my_blob, 3)) {
- char *s = my_blob + 3;
+ const char *s = my_blob + 3;
    char *d = my_blob;
    Trace(("trim BOM"));
    do {
@@ -329,7 +326,7 @@
    char *s;
    int y, x;
#if USE_WIDEC_SUPPORT
- char *last = my_vec[k] + (int) strlen(my_vec[k]);
+ const char *last = my_vec[k] + (int) strlen(my_vec[k]);
    wchar_t wch[2];
    size_t rc;
#endif state_unused
@@ -408,7 +405,7 @@

```

```

int
main(int argc, char *argv[])
{
- static const char *help[] =
+ static NCURSES_CONST char *help[] =
{
"Commands:",
" q,^Q,ESC - quit this program",
@@ -659,7 +656,7 @@
    beep();
    break;
}
- if (c >= KEY_MIN || (c > 0 && !isdigit(c))) {
+ if (c >= KEY_MIN || (c > 0 && !isdigit(UChar(c)))) {
    got_number = FALSE;
    value = 0;
}

```

Found in path(s):

* /ncurses-6.5-20241006-patch-gz/ncurses-6.5-20241006.patch

1.39 scanelf 1.3.8-r1

1.39.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses

are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish,
that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit
royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute
software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies
a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.40 utmps-lib 0.1.2.3-r2

1.40.1 Available under license :

Copyright (c) 2018-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.41 skalibs-lib 2.14.3.0-r0

1.41.1 Available under license :

Copyright (c) 2011-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.42 auto-sdk 1.1.0

1.42.1 Available under license :

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

// Collection is a type that implements the sort.Interface interface

// so that versions can be sorted.

```
type Collection []*Version
```

```
func (v Collection) Len() int {  
    return len(v)  
}
```

```
func (v Collection) Less(i, j int) bool {  
    return v[i].LessThan(v[j])  
}
```

```
func (v Collection) Swap(i, j int) {  
    v[i], v[j] = v[j], v[i]  
}
```

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

```
import (  
    "reflect"  
    "sort"  
    "testing"  
)
```

```
func TestCollection(t *testing.T) {  
    versionsRaw := []string{  
        "1.1.1",  
        "1.0",  
        "1.2",  
        "2",  
        "0.7.1",  
    }  
}
```

```
versions := make([]*Version, len(versionsRaw))  
for i, raw := range versionsRaw {  
    v, err := NewVersion(raw)  
    if err != nil {  
        t.Fatalf("err: %s", err)  
    }  
}
```

```
versions[i] = v
}
```

```
sort.Sort(Collection(versions))
```

```
actual := make([]string, len(versions))
for i, v := range versions {
    actual[i] = v.String()
}
```

```
expected := []string{
    "0.7.1",
    "1.0.0",
    "1.1.1",
    "1.2.0",
    "2.0.0",
}
```

```
if !reflect.DeepEqual(actual, expected) {
    t.Fatalf("bad: %#v", actual)
}
}
```

name: go-tests

on: [push]

env:

TEST_RESULTS: /tmp/test-results

jobs:

go-tests:

runs-on: ubuntu-latest

strategy:

matrix:

go-version: [1.15.3, 1.19]

steps:

- name: Setup go

uses: actions/setup-go@41dfa10bad2bb2ae585af6ee5bb4d7d973ad74ed # v5.1.0

with:

go-version: \${{ matrix.go-version }}

- name: Checkout code

uses: actions/checkout@11bd71901bbe5b1630ceea73d27597364c9af683 # v4.2.2

- name: Create test directory

run: |

```

mkdir -p ${ env.TEST_RESULTS }

- name: Download go modules
  run: go mod download

- name: Cache / restore go modules
  uses: actions/cache@1bd1e32a3bdc45362d1e726936510720a7c30a57 # v4.2.0
  with:
    path: |
      ~/go/pkg/mod
    key: ${ runner.os }-go-${ hashFiles('**/go.sum') }
    restore-keys: |
      ${ runner.os }-go-

# Check go fmt output because it does not report
non-zero when there are fmt changes
- name: Run gofmt
  run: |
    go fmt ./...
    files=$(go fmt ./...)
    if [ -n "$files" ]; then
      echo "The following file(s) do not conform to go fmt:"
      echo "$files"
      exit 1
    fi

# Install gotestsum with go get for 1.15.3; otherwise default to go install
- name: Install gotestsum
  run: |
    GTS="gotest.tools/gotestsum@v1.8.2"
    # We use the same error message prefix in either failure case, so just define it once here.
    ERROR="Failed to install $GTS"
    # First try to 'go install', if that fails try 'go get'...
    go install "$GTS" || go get "$GTS" || { echo "$ERROR: both 'go install' and 'go get' failed"; exit 1; }
    # Check that the gotestsum command was actually installed in the path...
    command -v gotestsum > /dev/null 2>&1 || { echo "$ERROR: gotestsum command not installed"; exit
1; }
    echo "OK: Command 'gotestsum' installed ($GTS)"

- name: Run go tests
  run: |
    PACKAGE_NAMES=$(go list ./...)
    gotestsum --format=short-verbose --junitfile $TEST_RESULTS/gotestsum-report.xml --
$PACKAGE_NAMES

# Save coverage report parts
- name: Upload and save artifacts
  uses: actions/upload-artifact@b4b15b8c7c6ac21ea08fcf65892d2ee8f75cf882 # v4.4.3

```


with:

name: Test Results

path: \${{ env.TEST_RESULTS }}

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Versioning Library for Go

![Build Status](https://github.com/hashicorp/go-version/actions/workflows/go-tests.yml/badge.svg)

![GoDoc](https://godoc.org/github.com/hashicorp/go-version?status.svg)(https://godoc.org/github.com/hashicorp/go-version)

go-version is a library for parsing versions and version constraints, and verifying versions against a set of constraints. go-version can sort a collection of versions properly, handles prerelease/beta versions, can increment versions, etc.

Versions used with go-version must follow [SemVer](http://semver.org/).

Installation and Usage

Package documentation can be found on
[GoDoc](http://godoc.org/github.com/hashicorp/go-version).

Installation can be done with a normal `go get`:

```

\$ go get github.com/hashicorp/go-version

```

Version Parsing and Comparison

```go

v1, err := version.NewVersion("1.2")

```

v2, err := version.NewVersion("1.5+metadata")

// Comparison example. There is also GreaterThan, Equal, and just
// a simple Compare
// that returns an int allowing easy >=, <=, etc.
if v1.LessThan(v2) {
 fmt.Printf("%s is less than %s", v1, v2)
}

```

#### #### Version Constraints

```

```go
v1, err := version.NewVersion("1.2")

// Constraints example.
constraints, err := version.NewConstraint(">= 1.0, < 1.4")
if constraints.Check(v1) {
    fmt.Printf("%s satisfies constraints %s", v1, constraints)
}

```

Version Sorting

```

```go
versionsRaw := []string{"1.1", "0.7.1", "1.4-beta", "1.4", "2"}
versions := make([]*version.Version, len(versionsRaw))
for i, raw := range versionsRaw {
 v, _ := version.NewVersion(raw)
 versions[i] = v
}

// After this, the versions are properly sorted
sort.Sort(version.Collection(versions))

```

#### ## Issues and Contributing

If you find an issue with this library, please report an issue. If you'd like, we welcome any contributions. Fork this library and submit a pull request.

The MIT License (MIT)

Copyright (c) 2014 Cenk Alt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to

use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011 The Snappy-Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2015, Dave Cheney <dave@cheney.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2015, Gengo, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Gengo, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009 The Go Authors.



Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

```
import (
 "encoding/json"
 "fmt"
 "reflect"
 "testing"
)
```

```
func TestNewVersion(t *testing.T) {
 cases := []struct {
 version string
 err bool
 }{
 {"", true},
 {"1.2.3", false},
 {"1.0", false},
 {"1", false},
 {"1.2.beta", true},
 }
```

```

{"1.21.beta", true},
{"foo", true},
{"1.2-5", false},
{"1.2-beta.5", false},
{"\n1.2", true},
{"1.2.0-x.Y.0+metadata", false},
{"1.2.0-x.Y.0+metadata-width-hyphen", false},
{"1.2.3-rc1-with-hyphen", false},
{"1.2.3.4", false},
{"1.2.0.4-x.Y.0+metadata", false},
{"1.2.0.4-x.Y.0+metadata-width-hyphen", false},
{"1.2.0-X-1.2.0+metadata~dist", false},
{"1.2.3.4-rc1-with-hyphen", false},
{"1.2.3.4", false},
{"v1.2.3", false},
{"foo1.2.3", true},
{"1.7rc2", false},
{"v1.7rc2", false},
{"1.0-", false},
}

```

```

for _, tc := range cases {
 _, err := NewVersion(tc.version)
 if tc.err && err == nil {
 t.Fatalf("expected error for version:
 %q", tc.version)
 } else if !tc.err && err != nil {
 t.Fatalf("error for version %q: %s", tc.version, err)
 }
}
}

```

```

func TestNewSemver(t *testing.T) {
 cases := []struct {
 version string
 err bool
 }{
 {"", true},
 {"1.2.3", false},
 {"1.0", false},
 {"1", false},
 {"1.2.beta", true},
 {"1.21.beta", true},
 {"foo", true},
 {"1.2-5", false},
 {"1.2-beta.5", false},
 {"\n1.2", true},
 {"1.2.0-x.Y.0+metadata", false},
 }
}

```

```

{"1.2.0-x.Y.0+metadata-width-hyphen", false},
{"1.2.3-rc1-with-hyphen", false},
{"1.2.3.4", false},
{"1.2.0.4-x.Y.0+metadata", false},
{"1.2.0.4-x.Y.0+metadata-width-hyphen", false},
{"1.2.0-X-1.2.0+metadata~dist", false},
{"1.2.3.4-rc1-with-hyphen", false},
{"1.2.3.4", false},
{"v1.2.3", false},
{"foo1.2.3", true},
{"1.7rc2", true},
{"v1.7rc2", true},
{"1.0-", true},
}

```

```

for _, tc := range cases {
 _, err := NewSemver(tc.version)
 if tc.err && err == nil {
 t.Fatalf("expected error for version: %q", tc.version)
 } else
 if !tc.err && err != nil {
 t.Fatalf("error for version %q: %s", tc.version, err)
 }
}
}

```

```

func TestCore(t *testing.T) {
 cases := []struct {
 v1 string
 v2 string
 }{
 {"1.2.3", "1.2.3"},
 {"2.3.4-alpha1", "2.3.4"},
 {"3.4.5alpha1", "3.4.5"},
 {"1.2.3-2", "1.2.3"},
 {"4.5.6-beta1+meta", "4.5.6"},
 {"5.6.7.1.2.3", "5.6.7"},
 }
}

```

```

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("error for version %q: %s", tc.v1, err)
 }
 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("error for version %q: %s", tc.v2, err)
 }
}

```

```

actual := v1.Core()
expected := v2

if !reflect.DeepEqual(actual, expected) {
 t.Fatalf("expected: %s\nactual: %s", expected, actual)
}
}
}

```

```

func TestVersionCompare(t *testing.T) {
 cases := []struct {
 v1 string
 v2 string
 expected int
 }{
 {"1.2.3", "1.4.5", -1},
 {"1.2-beta", "1.2-beta", 0},
 {"1.2", "1.1.4", 1},
 {"1.2", "1.2-beta", 1},
 {"1.2+foo", "1.2+beta", 0},
 {"v1.2", "v1.2-beta",
1},
 {"v1.2+foo", "v1.2+beta", 0},
 {"v1.2.3.4", "v1.2.3.4", 0},
 {"v1.2.0.0", "v1.2", 0},
 {"v1.2.0.0.1", "v1.2", 1},
 {"v1.2", "v1.2.0.0", 0},
 {"v1.2", "v1.2.0.0.1", -1},
 {"v1.2.0.0", "v1.2.0.0.1", -1},
 {"v1.2.3.0", "v1.2.3.4", -1},
 {"1.7rc2", "1.7rc1", 1},
 {"1.7rc2", "1.7", -1},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist", 1},
 }
}

```

```

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }
}

```

```

v2, err := NewVersion(tc.v2)
if err != nil {
 t.Fatalf("err: %s", err)
}

```

```

actual := v1.Compare(v2)

```

```

expected := tc.expected
if actual != expected {
 t.Fatalf(
 "%s <=> %s\nexpected: %d\nactual: %d",
 tc.v1, tc.v2,
 expected, actual)
}
}
}

func TestVersionCompare_versionAndSemver(t *testing.T) {
 cases := []struct {
 versionRaw string
 semverRaw string
 expected int
 }{
 {"0.0.2", "0.0.2", 0},
 {"1.0.2alpha", "1.0.2-alpha", 0},
 {"v1.2+foo", "v1.2+beta", 0},
 {"v1.2", "v1.2+meta", 0},
 {"1.2",
 "1.2-beta", 1},
 {"v1.2", "v1.2-beta", 1},
 {"1.2.3", "1.4.5", -1},
 {"v1.2", "v1.2.0.0.1", -1},
 {"v1.0.3-", "v1.0.3", -1},
 }

 for _, tc := range cases {
 ver, err := NewVersion(tc.versionRaw)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 semver, err := NewSemver(tc.semverRaw)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := ver.Compare(semver)
 if actual != tc.expected {
 t.Fatalf(
 "%s <=> %s\nexpected: %d\n actual: %d",
 tc.versionRaw, tc.semverRaw, tc.expected, actual,
)
 }
 }
}

```

```

func TestVersionEqual_nil(t *testing.T) {
 mustVersion := func(v string) *Version {
 ver, err := NewVersion(v)
 if err != nil {
 t.Fatal(err)
 }
 return ver
 }
 cases := []struct {
 leftVersion *Version
 rightVersion *Version
 expected bool
 }{
 {mustVersion("1.0.0"), nil, false},
 {nil, mustVersion("1.0.0"), false},
 {nil, nil, true},
 }

 for _, tc := range cases {
 given := tc.leftVersion.Equal(tc.rightVersion)
 if given != tc.expected {
 t.Fatalf("expected
Equal to nil to be %t", tc.expected)
 }
 }
}

```

```

func TestComparePreReleases(t *testing.T) {
 cases := []struct {
 v1 string
 v2 string
 expected int
 }{
 {"1.2-beta.2", "1.2-beta.2", 0},
 {"1.2-beta.1", "1.2-beta.2", -1},
 {"1.2-beta.2", "1.2-beta.11", -1},
 {"3.2-alpha.1", "3.2-alpha", 1},
 {"1.2-beta.2", "1.2-beta.1", 1},
 {"1.2-beta.11", "1.2-beta.2", 1},
 {"1.2-beta", "1.2-beta.3", -1},
 {"1.2-alpha", "1.2-beta.3", -1},
 {"1.2-beta", "1.2-alpha.3", 1},
 {"3.0-alpha.3", "3.0-rc.1", -1},
 {"3.0-alpha3", "3.0-rc1", -1},
 {"3.0-alpha.1", "3.0-alpha.beta", -1},
 {"5.4-alpha", "5.4-alpha.beta", 1},
 {"v1.2-beta.2", "v1.2-beta.2", 0},
 }
}

```

```

{"v1.2-beta.1", "v1.2-beta.2", -1},
{"v3.2-alpha.1", "v3.2-alpha", 1},
{"v3.2-rc.1-1-g123", "v3.2-rc.2", 1},
}

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual
:= v1.Compare(v2)
 expected := tc.expected
 if actual != expected {
 t.Fatalf(
 "%s <=> %s\nexpected: %d\nactual: %d",
 tc.v1, tc.v2,
 expected, actual)
 }
}

func TestVersionMetadata(t *testing.T) {
 cases := []struct {
 version string
 expected string
 }{
 {"1.2.3", ""},
 {"1.2-beta", ""},
 {"1.2.0-x.Y.0", ""},
 {"1.2.0-x.Y.0+metadata", "metadata"},
 {"1.2.0-metadata-1.2.0+metadata~dist", "metadata~dist"},
 }

 for _, tc := range cases {
 v, err := NewVersion(tc.version)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v.Metadata()
 expected := tc.expected

```

```

 if actual != expected {
 t.Fatalf("expected: %s\nactual: %s", expected, actual)
 }
}

func TestVersionPrerelease(t *testing.T) {
 cases := []struct {
 version string
 expected string
 }{
 {"1.2.3", ""},
 {"1.2-beta", "beta"},
 {"1.2.0-x.Y.0", "x.Y.0"},
 {"1.2.0-7.Y.0", "7.Y.0"},
 {"1.2.0-x.Y.0+metadata", "x.Y.0"},
 {"1.2.0-metadata-1.2.0+metadata~dist", "metadata-1.2.0"},
 {"17.03.0-ce",
"ce"}, // zero-padded fields
 }

 for _, tc := range cases {
 v, err := NewVersion(tc.version)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v.Prerelease()
 expected := tc.expected
 if actual != expected {
 t.Fatalf("expected: %s\nactual: %s", expected, actual)
 }
 }
}

func TestVersionSegments(t *testing.T) {
 cases := []struct {
 version string
 expected []int
 }{
 {"1.2.3", []int{1, 2, 3}},
 {"1.2-beta", []int{1, 2, 0}},
 {"1-x.Y.0", []int{1, 0, 0}},
 {"1.2.0-x.Y.0+metadata", []int{1, 2, 0}},
 {"1.2.0-metadata-1.2.0+metadata~dist", []int{1, 2, 0}},
 {"17.03.0-ce", []int{17, 3, 0}}, // zero-padded fields
 }
}

```



```

for _, tc := range cases {
 v, err := NewVersion(tc.version)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v.Segments()
 expected := tc.expected
 if !reflect.DeepEqual(actual, expected) {
 t.Fatalf("expected: %#v\nactual: %#v", expected, actual)
 }
}

func TestVersionSegments64(t *testing.T) {
 cases
 := []struct {
 version string
 expected []int64
 }{
 {"1.2.3", []int64{1, 2, 3}},
 {"1.2-beta", []int64{1, 2, 0}},
 {"1-x.Y.0", []int64{1, 0, 0}},
 {"1.2.0-x.Y.0+metadata", []int64{1, 2, 0}},
 {"1.4.9223372036854775807", []int64{1, 4, 9223372036854775807}},
 }

 for _, tc := range cases {
 v, err := NewVersion(tc.version)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v.Segments64()
 expected := tc.expected
 if !reflect.DeepEqual(actual, expected) {
 t.Fatalf("expected: %#v\nactual: %#v", expected, actual)
 }

 {
 expected := actual[0]
 actual[0]++
 actual = v.Segments64()
 if actual[0] != expected {
 t.Fatalf("Segments64 is mutable")
 }
 }
 }
}

```

```

}

func TestJsonMarshal(t *testing.T) {
 cases := []struct {
 version string
 err bool
 }{
 {"1.2.3", false},
 {"1.2.0-x.Y.0+metadata", false},
 {"1.2.0-x.Y.0+metadata-width-hyphen", false},
 {"1.2.3-rc1-with-hyphen", false},
 {"1.2.3.4", false},
 {"1.2.0.4-x.Y.0+metadata", false},
 {"1.2.0.4-x.Y.0+metadata-width-hyphen",
false},
 {"1.2.0-X-1.2.0+metadata~dist", false},
 {"1.2.3.4-rc1-with-hyphen", false},
 {"1.2.3.4", false},
 }

 for _, tc := range cases {
 v, err1 := NewVersion(tc.version)
 if err1 != nil {
 t.Fatalf("error for version %q: %s", tc.version, err1)
 }

 parsed, err2 := json.Marshal(v)
 if err2 != nil {
 t.Fatalf("error marshaling version %q: %s", tc.version, err2)
 }
 result := string(parsed)
 expected := fmt.Sprintf("%q", tc.version)
 if result != expected && !tc.err {
 t.Fatalf("Error marshaling unexpected marshaled content: result=%q expected=%q", result, expected)
 }
 }
}

func TestJsonUnmarshal(t *testing.T) {
 cases := []struct {
 version string
 err bool
 }{
 {"1.2.3", false},
 {"1.2.0-x.Y.0+metadata", false},
 {"1.2.0-x.Y.0+metadata-width-hyphen", false},
 {"1.2.3-rc1-with-hyphen", false},
 {"1.2.3.4", false},
 }

```

```

{"1.2.0.4-x.Y.0+metadata", false},
{"1.2.0.4-x.Y.0+metadata-width-hyphen", false},
{"1.2.0-X-1.2.0+metadata~dist",
false},
{"1.2.3.4-rc1-with-hyphen", false},
{"1.2.3.4", false},
}

for _, tc := range cases {
 expected, err1 := NewVersion(tc.version)
 if err1 != nil {
 t.Fatalf("err: %s", err1)
 }

 actual := &Version{}
 err2 := json.Unmarshal([]byte(fmt.Sprintf("%q", tc.version)), actual)
 if err2 != nil {
 t.Fatalf("error unmarshaling version: %s", err2)
 }
 if !reflect.DeepEqual(actual, expected) {
 t.Fatalf("error unmarshaling, unexpected object content: actual=%q expected=%q", actual, expected)
 }
}

func TestVersionString(t *testing.T) {
 cases := [][]string{
 {"1.2.3", "1.2.3"},
 {"1.2-beta", "1.2.0-beta"},
 {"1.2.0-x.Y.0", "1.2.0-x.Y.0"},
 {"1.2.0-x.Y.0+metadata", "1.2.0-x.Y.0+metadata"},
 {"1.2.0-metadata-1.2.0+metadata~dist", "1.2.0-metadata-1.2.0+metadata~dist"},
 {"17.03.0-ce", "17.3.0-ce"}, // zero-padded fields
 }

 for _, tc := range cases {
 v, err := NewVersion(tc[0])
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v.String()
 expected
 := tc[1]
 if actual != expected {
 t.Fatalf("expected: %s\nactual: %s", expected, actual)
 }
 if actual := v.Original(); actual != tc[0] {

```

```

 t.Fatalf("expected original: %q\nactual: %q", tc[0], actual)
}
}
}

```

```

func TestEqual(t *testing.T) {
 cases := []struct {
 v1 string
 v2 string
 expected bool
 }{
 {"1.2.3", "1.4.5", false},
 {"1.2-beta", "1.2-beta", true},
 {"1.2", "1.1.4", false},
 {"1.2", "1.2-beta", false},
 {"1.2+foo", "1.2+beta", true},
 {"v1.2", "v1.2-beta", false},
 {"v1.2+foo", "v1.2+beta", true},
 {"v1.2.3.4", "v1.2.3.4", true},
 {"v1.2.0.0", "v1.2", true},
 {"v1.2.0.0.1", "v1.2", false},
 {"v1.2", "v1.2.0.0", true},
 {"v1.2", "v1.2.0.0.1", false},
 {"v1.2.0.0", "v1.2.0.0.1", false},
 {"v1.2.3.0", "v1.2.3.4", false},
 {"1.7rc2", "1.7rc1", false},
 {"1.7rc2", "1.7", false},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist", false},
 }
}

```

```

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err:
%s", err)
 }
}

```

```

v2, err := NewVersion(tc.v2)
if err != nil {
 t.Fatalf("err: %s", err)
}

```

```

actual := v1.Equal(v2)
expected := tc.expected
if actual != expected {
 t.Fatalf(
 "%s <=> %s\nexpected: %t\nactual: %t",
 tc.v1, tc.v2,

```

```

 expected, actual)
}
}
}

func TestGreaterThan(t *testing.T) {
cases := []struct {
 v1 string
 v2 string
 expected bool
}{
 {"1.2.3", "1.4.5", false},
 {"1.2-beta", "1.2-beta", false},
 {"1.2", "1.1.4", true},
 {"1.2", "1.2-beta", true},
 {"1.2+foo", "1.2+beta", false},
 {"v1.2", "v1.2-beta", true},
 {"v1.2+foo", "v1.2+beta", false},
 {"v1.2.3.4", "v1.2.3.4", false},
 {"v1.2.0.0", "v1.2", false},
 {"v1.2.0.0.1", "v1.2", true},
 {"v1.2", "v1.2.0.0", false},
 {"v1.2", "v1.2.0.0.1", false},
 {"v1.2.0.0", "v1.2.0.0.1", false},
 {"v1.2.3.0", "v1.2.3.4", false},
 {"1.7rc2", "1.7rc1", true},
 {"1.7rc2", "1.7", false},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist", true},
}

for _, tc := range cases {
 v1,
 err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v1.GreaterThan(v2)
 expected := tc.expected
 if actual != expected {
 t.Fatalf(
 "%s > %s\nexpected: %t\nactual: %t",
 tc.v1, tc.v2,

```

```

 expected, actual)
 }
}
}

func TestLessThan(t *testing.T) {
cases := []struct {
 v1 string
 v2 string
 expected bool
}{
 {"1.2.3", "1.4.5", true},
 {"1.2-beta", "1.2-beta", false},
 {"1.2", "1.1.4", false},
 {"1.2", "1.2-beta", false},
 {"1.2+foo", "1.2+beta", false},
 {"v1.2", "v1.2-beta", false},
 {"v1.2+foo", "v1.2+beta", false},
 {"v1.2.3.4", "v1.2.3.4", false},
 {"v1.2.0.0", "v1.2", false},
 {"v1.2.0.0.1", "v1.2", false},
 {"v1.2", "v1.2.0.0", false},
 {"v1.2", "v1.2.0.0.1", true},
 {"v1.2.0.0", "v1.2.0.0.1", true},
 {"v1.2.3.0", "v1.2.3.4", true},
 {"1.7rc2", "1.7rc1", false},
 {"1.7rc2", "1.7", true},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist",
false},
}

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v1.LessThan(v2)
 expected := tc.expected
 if actual != expected {
 t.Fatalf(
 "%s < %s\nexpected: %t\nactual: %t",
 tc.v1, tc.v2,

```

```

 expected, actual)
}
}
}

func TestGreaterThanOrEqual(t *testing.T) {
cases := []struct {
 v1 string
 v2 string
 expected bool
}{
 {"1.2.3", "1.4.5", false},
 {"1.2-beta", "1.2-beta", true},
 {"1.2", "1.1.4", true},
 {"1.2", "1.2-beta", true},
 {"1.2+foo", "1.2+beta", true},
 {"v1.2", "v1.2-beta", true},
 {"v1.2+foo", "v1.2+beta", true},
 {"v1.2.3.4", "v1.2.3.4", true},
 {"v1.2.0.0", "v1.2", true},
 {"v1.2.0.0.1", "v1.2", true},
 {"v1.2", "v1.2.0.0", true},
 {"v1.2", "v1.2.0.0.1", false},
 {"v1.2.0.0", "v1.2.0.0.1", false},
 {"v1.2.3.0", "v1.2.3.4", false},
 {"1.7rc2", "1.7rc1", true},
 {"1.7rc2",
"1.7", false},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist", true},
}

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v1.GreaterThanOrEqual(v2)
 expected := tc.expected
 if actual != expected {
 t.Fatalf(
 "%s >= %s\nexpected: %t\nactual: %t",
 tc.v1, tc.v2,

```

```

 expected, actual)
}
}
}

func TestLessThanOrEqual(t *testing.T) {
cases := []struct {
 v1 string
 v2 string
 expected bool
}{
 {"1.2.3", "1.4.5", true},
 {"1.2-beta", "1.2-beta", true},
 {"1.2", "1.1.4", false},
 {"1.2", "1.2-beta", false},
 {"1.2+foo", "1.2+beta", true},
 {"v1.2", "v1.2-beta", false},
 {"v1.2+foo", "v1.2+beta", true},
 {"v1.2.3.4", "v1.2.3.4", true},
 {"v1.2.0.0", "v1.2", true},
 {"v1.2.0.0.1", "v1.2", false},
 {"v1.2", "v1.2.0.0", true},
 {"v1.2", "v1.2.0.0.1", true},
 {"v1.2.0.0", "v1.2.0.0.1", true},
 {"v1.2.3.0",
"v1.2.3.4", true},
 {"1.7rc2", "1.7rc1", false},
 {"1.7rc2", "1.7", true},
 {"1.2.0", "1.2.0-X-1.2.0+metadata~dist", false},
}

for _, tc := range cases {
 v1, err := NewVersion(tc.v1)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 v2, err := NewVersion(tc.v2)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := v1.LessThanOrEqual(v2)
 expected := tc.expected
 if actual != expected {
 t.Fatalf(
 "%s <= %s\nexpected: %t\nactual: %t",
 tc.v1, tc.v2,

```



```
 expected, actual)
 }
}
}
```

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).  
MIT License

Copyright (c) 2016 json-iterator

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2011, Open Knowledge Foundation Ltd.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2014 HashiCorp, Inc.

Mozilla Public License, version 2.0

## 1. Definitions

### 1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

### 1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

### 1.3. Contribution

means Covered Software of a particular Contributor.

### 1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

### 1.5. Incompatible With Secondary Licenses

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

### 1.6. Executable Form

means any form of the work other than Source Code Form.

#### 1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. License

means this document.

#### 1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. Modifications

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus

claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. Source Code Form

means the form of the work preferred for making modifications.

#### 1.14. You (or Your)

means an individual or a legal entity exercising rights under this

License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its

Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

#### 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

#### 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

#### 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

#### 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

### 3. Responsibilities

#### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

#### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form,

as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable

Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates

Your grants, and (b) on an ongoing basis,

if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance.

Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the

event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or



correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

## 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

## 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

## 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

### Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

Copyright (c) 2016 Caleb Spare

### MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH

THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.7.0 (May 24, 2024)

#### ENHANCEMENTS:

- Remove `reflect` dependency ([#91](https://github.com/hashicorp/go-version/pull/91))
- Implement the `database/sql.Scanner` and `database/sql/driver.Value` interfaces for `Version` ([#133](https://github.com/hashicorp/go-version/pull/133))

#### INTERNAL:

- [COMPLIANCE] Add Copyright and License Headers ([#115](https://github.com/hashicorp/go-version/pull/115))
- [COMPLIANCE] Update MPL-2.0 LICENSE ([#105](https://github.com/hashicorp/go-version/pull/105))
- Bump actions/cache from 3.0.11 to 3.2.5 ([#116](https://github.com/hashicorp/go-version/pull/116))
- Bump actions/checkout from 3.2.0 to 3.3.0 ([#111](https://github.com/hashicorp/go-version/pull/111))
- Bump actions/upload-artifact from 3.1.1 to 3.1.2 ([#112](https://github.com/hashicorp/go-version/pull/112))
- GHA Migration ([#103](https://github.com/hashicorp/go-version/pull/103))
- github: Pin external GitHub Actions to hashes ([#107](https://github.com/hashicorp/go-version/pull/107))
- SEC-090:  
Automated trusted workflow pinning (2023-04-05) ([#124](https://github.com/hashicorp/go-version/pull/124))
- update readme ([#104](https://github.com/hashicorp/go-version/pull/104))

# 1.6.0 (June 28, 2022)

#### FEATURES:

- Add `Prerelease` function to `Constraint` to return true if the version includes a prerelease field ([#100](https://github.com/hashicorp/go-version/pull/100))

# 1.5.0 (May 18, 2022)

#### FEATURES:

- Use ``encoding`` ``TextMarshaler`` & ``TextUnmarshaler`` instead of JSON equivalents ([#95](https://github.com/hashicorp/go-version/pull/95))
- Add JSON handlers to allow parsing from/to JSON ([#93](https://github.com/hashicorp/go-version/pull/93))

# 1.4.0 (January 5, 2022)

#### FEATURES:

- Introduce ``MustConstraints()`` ([#87](https://github.com/hashicorp/go-version/pull/87))
- ``Constraints``: Introduce ``Equals()`` and ``sort.Interface`` methods ([#88](https://github.com/hashicorp/go-version/pull/88))

# 1.3.0 (March 31, 2021)

Please note that CHANGELOG.md does not exist in the source code prior to this release.

#### FEATURES:

- Add ``Core`` function to return a version without prerelease or metadata ([#85](https://github.com/hashicorp/go-version/pull/85))

# 1.2.1 (June 17, 2020)

#### BUG FIXES:

- Prevent ``Version.Equal`` method from panicking on ``nil`` encounter ([#73](https://github.com/hashicorp/go-version/pull/73))

# 1.2.0 (April 23, 2019)

#### FEATURES:

- Add ``GreaterThanOrEqual`` and ``LessThanOrEqual`` helper methods ([#53](https://github.com/hashicorp/go-version/pull/53))

# 1.1.0 (Jan 07, 2019)

#### FEATURES:

- Add ``NewSemver`` constructor ([#45](https://github.com/hashicorp/go-version/pull/45))

# 1.0.0 (August 24, 2018)

Initial release.

procf provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at SoundCloud Ltd. (<http://soundcloud.com/>).

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding  
those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional attribution notices cannot be construed  
as modifying the License.
- You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions
- for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a



file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2017-2021 Uber Technologies, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

```
import (
 "fmt"
 "regexp"
 "sort"
 "strings"
)
```

```

// Constraint represents a single constraint for a version, such as
// ">= 1.0".
type Constraint struct {
 f constraintFunc
 op operator
 check *Version
 original string
}

func (c *Constraint) Equals(con *Constraint) bool {
 return c.op == con.op && c.check.Equal(con.check)
}

// Constraints is a slice of constraints. We make a custom type so that
// we can add methods to it.
type Constraints []*Constraint

type constraintFunc func(v, c *Version) bool

var constraintOperators map[string]constraintOperation

type constraintOperation struct {
 op operator
 f constraintFunc
}

var constraintRegexp *regexp.Regexp

func init() {
 constraintOperators = map[string]constraintOperation{
 "": {op: equal, f: constraintEqual},
 "=": {op: equal, f: constraintEqual},
 "!=": {op: notEqual, f: constraintNotEqual},
 ">": {op: greaterThan, f: constraintGreaterThan},
 "<": {
 op: lessThan, f: constraintLessThan},
 ">=": {op: greaterThanEqual, f: constraintGreaterThanOrEqual},
 "<=": {op: lessThanEqual, f: constraintLessThanOrEqual},
 "~>": {op: pessimistic, f: constraintPessimistic},
 }

 ops := make([]string, 0, len(constraintOperators))
 for k := range constraintOperators {
 ops = append(ops, regexp.QuoteMeta(k))
 }

 constraintRegexp = regexp.MustCompile(fmt.Sprintf(

```

```

 `^\s*(%s)\s*(%s)\s*$`,
 strings.Join(ops, "|"),
 VersionRegexpRaw))
}

// NewConstraint will parse one or more constraints from the given
// constraint string. The string must be a comma-separated list of
// constraints.
func NewConstraint(v string) (Constraints, error) {
 vs := strings.Split(v, ",")
 result := make([]*Constraint, len(vs))
 for i, single := range vs {
 c, err := parseSingle(single)
 if err != nil {
 return nil, err
 }

 result[i] = c
 }

 return Constraints(result), nil
}

// MustConstraints is a helper that wraps a call to a function
//
// returning (Constraints, error) and panics if error is non-nil.
func MustConstraints(c Constraints, err error) Constraints {
 if err != nil {
 panic(err)
 }

 return c
}

// Check tests if a version satisfies all the constraints.
func (cs Constraints) Check(v *Version) bool {
 for _, c := range cs {
 if !c.Check(v) {
 return false
 }
 }

 return true
}

// Equals compares Constraints with other Constraints
// for equality. This may not represent logical equivalence
// of compared constraints.

```

```

// e.g. even though '>0.1,>0.2' is logically equivalent
// to '>0.2' it is *NOT* treated as equal.
//
// Missing operator is treated as equal to '=', whitespaces
// are ignored and constraints are sorted before comparison.
func (cs Constraints) Equals(c Constraints) bool {
 if len(cs) != len(c) {
 return false
 }

 // make copies to retain order of the original slices
 left := make(Constraints, len(cs))
 copy(left, cs)
 sort.Stable(left)
 right := make(Constraints, len(c))
 copy(right, c)
 sort.Stable(right)

 //
 compare sorted slices
 for i, con := range left {
 if !con.Equals(right[i]) {
 return false
 }
 }

 return true
}

func (cs Constraints) Len() int {
 return len(cs)
}

func (cs Constraints) Less(i, j int) bool {
 if cs[i].op < cs[j].op {
 return true
 }
 if cs[i].op > cs[j].op {
 return false
 }

 return cs[i].check.LessThan(cs[j].check)
}

func (cs Constraints) Swap(i, j int) {
 cs[i], cs[j] = cs[j], cs[i]
}

```

```

// Returns the string format of the constraints
func (cs Constraints) String() string {
 csStr := make([]string, len(cs))
 for i, c := range cs {
 csStr[i] = c.String()
 }

 return strings.Join(csStr, ",")
}

// Check tests if a constraint is validated by the given version.
func (c *Constraint) Check(v *Version) bool {
 return c.f(v, c.check)
}

// Prerelease returns true if the version underlying this constraint
// contains a prerelease field.
func (c *Constraint) Prerelease() bool {
 return len(c.check.Prerelease()) > 0
}

func (c *Constraint) String()
string {
 return c.original
}

func parseSingle(v string) (*Constraint, error) {
 matches := constraintRegexp.FindStringSubmatch(v)
 if matches == nil {
 return nil, fmt.Errorf("Malformed constraint: %s", v)
 }

 check, err := NewVersion(matches[2])
 if err != nil {
 return nil, err
 }

 cop := constraintOperators[matches[1]]

 return &Constraint{
 f: cop.f,
 op: cop.op,
 check: check,
 original: v,
 }, nil
}

func prereleaseCheck(v, c *Version) bool {

```

```

switch vPre, cPre := v.Prerelease() != "", c.Prerelease() != ""; {
case cPre && vPre:
 // A constraint with a pre-release can only match a pre-release version
 // with the same base segments.
 return v.equalSegments(c)

case !cPre && vPre:
 // A constraint without a pre-release can only match a version without a
 // pre-release.
 return false

case cPre && !vPre:
 // OK, except with the pessimistic operator
case !cPre && !vPre:
 // OK
}
return true
}

//-----
//
// Constraint functions
//-----

type operator rune

const (
 equal operator = '='
 notEqual operator = "
 greaterThan operator = '>'
 lessThan operator = '<'
 greaterThanEqual operator = "
 lessThanEqual operator = "
 pessimistic operator = '~'
)

func constraintEqual(v, c *Version) bool {
 return v.Equal(c)
}

func constraintNotEqual(v, c *Version) bool {
 return !v.Equal(c)
}

func constraintGreaterThan(v, c *Version) bool {
 return prereleaseCheck(v, c) && v.Compare(c) == 1
}

```

```

func constraintLessThan(v, c *Version) bool {
 return prereleaseCheck(v, c) && v.Compare(c) == -1
}

func constraintGreaterThanEqual(v, c *Version) bool {
 return prereleaseCheck(v, c) && v.Compare(c) >= 0
}

func constraintLessThanEqual(v, c *Version) bool {
 return prereleaseCheck(v, c) && v.Compare(c) <= 0
}

func constraintPessimistic(v, c *Version) bool {
 // Using a pessimistic
 // constraint with a pre-release, restricts versions to pre-releases
 if !prereleaseCheck(v, c) || (c.Prerelease() != "" && v.Prerelease() == "") {
 return false
 }

 // If the version being checked is naturally less than the constraint, then there
 // is no way for the version to be valid against the constraint
 if v.LessThan(c) {
 return false
 }

 // We'll use this more than once, so grab the length now so it's a little cleaner
 // to write the later checks
 cs := len(c.segments)

 // If the version being checked has less specificity than the constraint, then there
 // is no way for the version to be valid against the constraint
 if cs > len(v.segments) {
 return false
 }

 // Check the segments in the constraint against those in the version. If the version
 // being checked, at any point, does not have the same values in each index of the
 // constraints segments, then it cannot be valid against the constraint.
 for i := 0; i < cs-1; i++ {
 if v.segments[i] != c.segments[i]
 {
 return false
 }
 }

 // Check the last part of the segment in the constraint. If the version segment at
 // this index is less than the constraints segment at this index, then it cannot
 // be valid against the constraint
 if c.segments[cs-1] > v.segments[cs-1] {

```

```

return false
}

// If nothing has rejected the version by now, it's valid
return true
}
version: 2
updates:
- package-ecosystem: "gomod"
 directory: "/"
 schedule:
 interval: "daily"
 labels: ["dependencies"]

- package-ecosystem: github-actions
 directory: /
 schedule:
 interval: monthly
 labels:
 - dependencies
 # only update HashiCorp actions, external actions managed by TSCCR
 allow:
 - dependency-name: hashicorp/*
 groups:
 github-actions-breaking:
 update-types:
 - major
 github-actions-backward-compatible:
 update-types:
 - minor
 - patch

```

Common libraries shared by Prometheus Go components.

Copyright 2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).

MIT License

Copyright (c) 2017 Nathan Sweet

Copyright (c) 2018, 2019 Cloudflare

Copyright (c) 2019 Authors of Cilium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:



The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

module github.com/hashicorp/go-version

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 The Go Authors. All rights reserved.

Copyright (c) 2019 Klaus Post. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----

Files: gzhttp/\*

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files;  
and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally

appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall

be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016-2017 The New York Times Company

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----

Files: s2/cmd/internal/readahead/\*

The MIT License  
(MIT)

Copyright (c) 2015 Klaus Post

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
Files: snappy/\*

Files: internal/snapref/\*

Copyright (c) 2011 The Snappy-Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED

TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----

Files: s2/cmd/internal/filepathx/\*

Copyright 2016 The filepathx Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

```
import (
 "fmt"
 "reflect"
 "sort"
 "testing"
)
```

```
func TestNewConstraint(t *testing.T) {
 cases := []struct {
 input string
```



```

count int
err bool
}{
{">= 1.2", 1, false},
{"1.0", 1, false},
{">= 1.x", 0, true},
{">= 1.2, < 1.0", 2, false},

// Out of bounds
{"11387778780781445675529500000000000000000000", 0, true},
}

for _, tc := range cases {
v, err := NewConstraint(tc.input)
if tc.err && err == nil {
t.Fatalf("expected error for input: %s", tc.input)
} else if !tc.err && err != nil {
t.Fatalf("error for input %s: %s", tc.input, err)
}

if len(v) != tc.count {
t.Fatalf("input: %s\nexpected len: %d\nactual: %d",
tc.input, tc.count, len(v))
}
}
}

func TestConstraintCheck(t *testing.T) {
cases := []struct {
constraint string
version string
check bool
}{
{">= 1.0, < 1.2", "1.1.5", true},
{"< 1.0, < 1.2", "1.1.5", false},
{"= 1.0", "1.1.5",
false},
{"= 1.0", "1.0.0", true},
{"1.0", "1.0.0", true},
{"~> 1.0", "2.0", false},
{"~> 1.0", "1.1", true},
{"~> 1.0", "1.2.3", true},
{"~> 1.0.0", "1.2.3", false},
{"~> 1.0.0", "1.0.7", true},
{"~> 1.0.0", "1.1.0", false},
{"~> 1.0.7", "1.0.4", false},
{"~> 1.0.7", "1.0.7", true},
{"~> 1.0.7", "1.0.8", true},

```

```

{"~> 1.0.7", "1.0.7.5", true},
{"~> 1.0.7", "1.0.6.99", false},
{"~> 1.0.7", "1.0.8.0", true},
{"~> 1.0.9.5", "1.0.9.5", true},
{"~> 1.0.9.5", "1.0.9.4", false},
{"~> 1.0.9.5", "1.0.9.6", true},
{"~> 1.0.9.5", "1.0.9.5.0", true},
{"~> 1.0.9.5", "1.0.9.5.1", true},
{"~> 2.0", "2.1.0-beta", false},
{"~> 2.1.0-a", "2.2.0", false},
{"~> 2.1.0-a", "2.1.0", false},
{"~> 2.1.0-a", "2.1.0-beta", true},
{"~> 2.1.0-a", "2.2.0-alpha", false},
{"> 2.0", "2.1.0-beta", false},
{">= 2.1.0-a", "2.1.0-beta", true},
{">= 2.1.0-a", "2.1.1-beta", false},
{">= 2.0.0", "2.1.0-beta", false},
{">= 2.1.0-a", "2.1.1", true},
{">= 2.1.0-a", "2.1.1-beta",
false},
{">= 2.1.0-a", "2.1.0", true},
{"<= 2.1.0-a", "2.0.0", true},
}

```

```

for _, tc := range cases {
 c, err := NewConstraint(tc.constraint)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

```

```

 v, err := NewVersion(tc.version)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

```

```

 actual := c.Check(v)
 expected := tc.check
 if actual != expected {
 t.Fatalf("Version: %s\nConstraint: %s\nExpected: %#v",
 tc.version, tc.constraint, expected)
 }
}
}

```

```

func TestConstraintPrerelease(t *testing.T) {
 cases := []struct {
 constraint string
 prerelease bool
 }

```

```

}{
 {"= 1.0", false},
 {"= 1.0-beta", true},
 {"~> 2.1.0", false},
 {"~> 2.1.0-dev", true},
 {"> 2.0", false},
 {">= 2.1.0-a", true},
}

for _, tc := range cases {
 c, err := parseSingle(tc.constraint)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := c.Prerelease()
 expected := tc.prerelease
 if actual != expected {
 t.Fatalf("Constraint: %s\nExpected: %#v",
 tc.constraint, expected)
 }
}

func
TestConstraintEqual(t *testing.T) {
 cases := []struct {
 leftConstraint string
 rightConstraint string
 expectedEqual bool
 }{
 {
 "0.0.1",
 "0.0.1",
 true,
 },
 { // whitespaces
 " 0.0.1 ",
 "0.0.1",
 true,
 },
 { // equal op implied
 "=0.0.1 ",
 "0.0.1",
 true,
 },
 { // version difference
 "=0.0.1",

```

```

 "=0.0.2",
 false,
 },
 { // operator difference
 ">0.0.1",
 "=0.0.1",
 false,
 },
 { // different order
 ">0.1.0, <=1.0.0",
 "<=1.0.0, >0.1.0",
 true,
 },
}

for _, tc := range cases {
 leftCon, err := NewConstraint(tc.leftConstraint)
 if err != nil {
 t.Fatalf("err: %s", err)
 }
 rightCon, err := NewConstraint(tc.rightConstraint)
 if err != nil {
 t.Fatalf("err: %s", err)
 }

 actual := leftCon.Equals(rightCon)
 if actual != tc.expectedEqual {
 t.Fatalf("Constraints: %s vs %s\nExpected: %t\nActual: %t",
 tc.leftConstraint, tc.rightConstraint, tc.expectedEqual, actual)
 }
}

func TestConstraint_sort(t
*testing.T) {
 cases := []struct {
 constraint string
 expectedConstraints string
 }{
 {
 ">= 0.1.0,< 1.12",
 "< 1.12,>= 0.1.0",
 },
 {
 "< 1.12,>= 0.1.0",
 "< 1.12,>= 0.1.0",
 },
 }
}

```

```

"< 1.12,>= 0.1.0,0.2.0",
"< 1.12,0.2.0,>= 0.1.0",
},
{
">1.0,>0.1.0,>0.3.0,>0.2.0",
">0.1.0,>0.2.0,>0.3.0,>1.0",
},
}

for i, tc := range cases {
t.Run(fmt.Sprintf("%d", i), func(t *testing.T) {
c, err := NewConstraint(tc.constraint)
if err != nil {
t.Fatalf("err: %s", err)
}

sort.Sort(c)

actual := c.String()

if !reflect.DeepEqual(actual, tc.expectedConstraints) {
t.Fatalf("unexpected order\nexpected: %#v\nactual: %#v",
tc.expectedConstraints, actual)
}
})
}

func TestConstraintsString(t *testing.T) {
cases := []struct {
constraint string
result string
}{
{">= 1.0, < 1.2", ""},
{"~> 1.0.7", ""},
}

for _, tc := range cases {
c, err := NewConstraint(tc.constraint)
if err != nil {
t.Fatalf("err:
%s", err)
}

actual := c.String()
expected := tc.result
if expected == "" {
expected = tc.constraint

```

```

}

if actual != expected {
 t.Fatalf("Constraint: %s\nExpected: %#v\nActual: %s",
 tc.constraint, expected, actual)
}
}
}

```

Prometheus instrumentation library for Go applications  
 Copyright 2012-2015 The Prometheus Authors

This product includes software developed at  
 SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>  
<https://github.com/beorn7/perks>  
 Copyright 2013-2015 Blake Mizerany, Björn Rabenstein  
 See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format  
<http://github.com/golang/protobuf/>  
 Copyright 2010 The Go Authors  
 See source code for license details.

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or



documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2013, The GoGo Authors. All rights reserved.

Protocol Buffers for Go with Gadgets

Go support for Protocol Buffers - Google's data interchange format

Copyright 2010 The Go Authors. All rights reserved.

<https://github.com/golang/protobuf>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Valid-License-Identifier: BSD-2-Clause

SPDX-URL: <https://spdx.org/licenses/BSD-2-Clause.html>

Usage-Guide:

To use the BSD 2-clause "Simplified" License put the following SPDX tag/value pair into a comment according to the placement guidelines in the licensing rules documentation:

SPDX-License-Identifier: BSD-2-Clause

License-Text:

Copyright (c) 2015 The Libbpf Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package version

```
import (
 "bytes"
 "database/sql/driver"
 "fmt"
 "regexp"
```

```

"strconv"
"strings"
)

// The compiled regular expression used to test the validity of a version.
var (
 versionRegexp *regexp.Regexp
 semverRegexp *regexp.Regexp
)

// The raw regular expression string used for testing the validity
// of a version.
const (
 VersionRegexpRaw string = `v?([0-9]+(\.[0-9]+)*)` +
 `(-([0-9]+[0-9A-Za-z\~]*\.[0-9A-Za-z\~]*)|(-?([A-Za-z\~]+[0-9A-Za-z\~]*\.[0-9A-Za-z\~]+)*)))?` +
 `(\+([0-9A-Za-z\~]+(\.[0-9A-Za-z\~]+)*))?'` +
 `?`

 // SemverRegexpRaw requires a separator between version and prerelease
 SemverRegexpRaw string = `v?([0-9]+(\.[0-9]+)*)` +
 `(-([0-9]+[0-9A-Za-z\~]*\.[0-9A-Za-z\~]*)|(-([A-Za-z\~]+[0-9A-Za-z\~]*\.[0-9A-Za-z\~]+)*)))?` +
 `(\+([0-9A-Za-z\~]+(\.[0-9A-Za-z\~]+)*))?'` +
 `?`
)

// Version represents a single version.
type Version struct {
 metadata string
 pre
 string
 segments []int64
 si int
 original string
}

func init() {
 versionRegexp = regexp.MustCompile("^" + VersionRegexpRaw + "$")
 semverRegexp = regexp.MustCompile("^" + SemverRegexpRaw + "$")
}

// NewVersion parses the given version and returns a new
// Version.
func NewVersion(v string) (*Version, error) {
 return newVersion(v, versionRegexp)
}

// NewSemver parses the given version and returns a new
// Version that adheres strictly to SemVer specs

```

```

// https://semver.org/
func NewSemver(v string) (*Version, error) {
 return newVersion(v, semverRegexp)
}

func newVersion(v string, pattern *regexp.Regexp) (*Version, error) {
 matches := pattern.FindStringSubmatch(v)
 if matches == nil {
 return nil, fmt.Errorf("Malformed version: %s", v)
 }
 segmentsStr := strings.Split(matches[1], ".")
 segments := make([]int64, len(segmentsStr))
 for i, str := range segmentsStr {
 val, err := strconv.ParseInt(str, 10, 64)
 if err != nil {
 return nil, fmt.Errorf(
 "Error parsing version:
%s", err)
 }

 segments[i] = val
 }

 // Even though we could support more than three segments, if we
 // got less than three, pad it with 0s. This is to cover the basic
 // default usecase of semver, which is MAJOR.MINOR.PATCH at the minimum
 for i := len(segments); i < 3; i++ {
 segments = append(segments, 0)
 }

 pre := matches[7]
 if pre == "" {
 pre = matches[4]
 }

 return &Version{
 metadata: matches[10],
 pre: pre,
 segments: segments,
 si: len(segmentsStr),
 original: v,
 }, nil
}

// Must is a helper that wraps a call to a function returning (*Version, error)
// and panics if error is non-nil.
func Must(v *Version, err error) *Version {
 if err != nil {

```

```

 panic(err)
}

return v
}

// Compare compares this version to another version. This
// returns -1, 0, or 1 if this version is smaller, equal,
// or larger than the other version, respectively.
//
// If you want boolean results, use the LessThan, Equal,
// GreaterThan, GreaterThanOrEqualTo or LessThanOrEqualTo
// methods.
func (v *Version) Compare(other *Version) int {
 // A quick, efficient equality check
 if v.String() == other.String() {
 return 0
 }

 // If the segments are the same, we must compare on prerelease info
 if v.equalSegments(other) {
 preSelf := v.Prerelease()
 preOther := other.Prerelease()
 if preSelf == "" && preOther == "" {
 return 0
 }
 if preSelf == "" {
 return 1
 }
 if preOther == "" {
 return -1
 }
 }

 return comparePrereleases(preSelf, preOther)
}

segmentsSelf := v.Segments64()
segmentsOther := other.Segments64()
// Get the highest specificity (hS), or if they're equal, just use segmentSelf length
lenSelf := len(segmentsSelf)
lenOther := len(segmentsOther)
hS := lenSelf
if lenSelf < lenOther {
 hS = lenOther
}
// Compare the segments
// Because a constraint could have more/less specificity than the version it's
// checking, we need to account for a lopsided or jagged comparison

```

```

for i := 0; i < hS; i++ {
 if i > lenSelf-1 {
 //
 This means Self had the lower specificity
 // Check to see if the remaining segments in Other are all zeros
 if !allZero(segmentsOther[i:]) {
 // if not, it means that Other has to be greater than Self
 return -1
 }
 break
 } else if i > lenOther-1 {
 // this means Other had the lower specificity
 // Check to see if the remaining segments in Self are all zeros -
 if !allZero(segmentsSelf[i:]) {
 // if not, it means that Self has to be greater than Other
 return 1
 }
 break
 }
 lhs := segmentsSelf[i]
 rhs := segmentsOther[i]
 if lhs == rhs {
 continue
 } else if lhs < rhs {
 return -1
 }
 // Otherwis, rhs was > lhs, they're not equal
 return 1
}

// if we got this far, they're equal
return 0
}

func (v *Version) equalSegments(other *Version) bool {
 segmentsSelf := v.Segments64()
 segmentsOther := other.Segments64()

 if len(segmentsSelf) != len(segmentsOther) {
 return false
 }
 for i, v := range segmentsSelf {
 if v != segmentsOther[i]
 {
 return false
 }
 }
 return true
}

```



```

}

func allZero(segs []int64) bool {
 for _, s := range segs {
 if s != 0 {
 return false
 }
 }
 return true
}

func comparePart(preSelf string, preOther string) int {
 if preSelf == preOther {
 return 0
 }

 var selfInt int64
 selfNumeric := true
 selfInt, err := strconv.ParseInt(preSelf, 10, 64)
 if err != nil {
 selfNumeric = false
 }

 var otherInt int64
 otherNumeric := true
 otherInt, err = strconv.ParseInt(preOther, 10, 64)
 if err != nil {
 otherNumeric = false
 }

 // if a part is empty, we use the other to decide
 if preSelf == "" {
 if otherNumeric {
 return -1
 }
 return 1
 }

 if preOther == "" {
 if selfNumeric {
 return 1
 }
 return -1
 }

 if selfNumeric && !otherNumeric {
 return -1
 } else if !selfNumeric && otherNumeric {

```

```

 return 1
} else if !selfNumeric && !otherNumeric && preSelf > preOther {
 return 1
} else if selfInt > otherInt {
 return 1
}

return -1
}

func comparePrereleases(v string, other string)
int {
 // the same pre release!
 if v == other {
 return 0
 }

 // split both pre releases for analyse their parts
 selfPreReleaseMeta := strings.Split(v, ".")
 otherPreReleaseMeta := strings.Split(other, ".")

 selfPreReleaseLen := len(selfPreReleaseMeta)
 otherPreReleaseLen := len(otherPreReleaseMeta)

 biggestLen := otherPreReleaseLen
 if selfPreReleaseLen > otherPreReleaseLen {
 biggestLen = selfPreReleaseLen
 }

 // loop for parts to find the first difference
 for i := 0; i < biggestLen; i = i + 1 {
 partSelfPre := ""
 if i < selfPreReleaseLen {
 partSelfPre = selfPreReleaseMeta[i]
 }

 partOtherPre := ""
 if i < otherPreReleaseLen {
 partOtherPre = otherPreReleaseMeta[i]
 }

 compare := comparePart(partSelfPre, partOtherPre)
 // if parts are equals, continue the loop
 if compare != 0 {
 return compare
 }
 }
}

```

```

return 0
}

// Core returns a new version constructed from only the MAJOR.MINOR.PATCH
// segments of the version, without prerelease or metadata.
func
(v *Version) Core() *Version {
 segments := v.Segments64()
 segmentsOnly := fmt.Sprintf("%d.%d.%d", segments[0], segments[1], segments[2])
 return Must(NewVersion(segmentsOnly))
}

// Equal tests if two versions are equal.
func (v *Version) Equal(o *Version) bool {
 if v == nil || o == nil {
 return v == o
 }

 return v.Compare(o) == 0
}

// GreaterThan tests if this version is greater than another version.
func (v *Version) GreaterThan(o *Version) bool {
 return v.Compare(o) > 0
}

// GreaterThanOrEqual tests if this version is greater than or equal to another version.
func (v *Version) GreaterThanOrEqual(o *Version) bool {
 return v.Compare(o) >= 0
}

// LessThan tests if this version is less than another version.
func (v *Version) LessThan(o *Version) bool {
 return v.Compare(o) < 0
}

// LessThanOrEqual tests if this version is less than or equal to another version.
func (v *Version) LessThanOrEqual(o *Version) bool {
 return v.Compare(o) <= 0
}

// Metadata returns any metadata
// that was part of the version
// string.
//
// Metadata is anything that comes after the "+" in the version.
// For example, with "1.2.3+beta", the metadata is "beta".
func (v *Version) Metadata() string {

```

```

return v.metadata
}

// Prerelease returns any prerelease data that is part of the version,
// or blank if there is no prerelease data.
//
// Prerelease information is anything that comes after the "-" in the
// version (but before any metadata). For example, with "1.2.3-beta",
// the prerelease information is "beta".
func (v *Version) Prerelease() string {
 return v.pre
}

// Segments returns the numeric segments of the version as a slice of ints.
//
// This excludes any metadata or pre-release information. For example,
// for a version "1.2.3-beta", segments will return a slice of
// 1, 2, 3.
func (v *Version) Segments() []int {
 segmentSlice := make([]int, len(v.segments))
 for i, v := range v.segments {
 segmentSlice[i] = int(v)
 }
 return segmentSlice
}

// Segments64 returns the
// numeric segments of the version as a slice of int64s.
//
// This excludes any metadata or pre-release information. For example,
// for a version "1.2.3-beta", segments will return a slice of
// 1, 2, 3.
func (v *Version) Segments64() []int64 {
 result := make([]int64, len(v.segments))
 copy(result, v.segments)
 return result
}

// String returns the full version string included pre-release
// and metadata information.
//
// This value is rebuilt according to the parsed segments and other
// information. Therefore, ambiguities in the version string such as
// prefixed zeroes (1.04.0 => 1.4.0), `v` prefix (v1.0.0 => 1.0.0), and
// missing parts (1.0 => 1.0.0) will be made into a canonicalized form
// as shown in the parenthesized examples.
func (v *Version) String() string {
 var buf bytes.Buffer

```

```

fmtParts := make([]string, len(v.segments))
for i, s := range v.segments {
 // We can ignore err here since we've pre-parsed the values in segments
 str := strconv.FormatInt(s, 10)
 fmtParts[i]
 = str
}
fmt.Fprintf(&buf, strings.Join(fmtParts, "."))
if v.pre != "" {
 fmt.Fprintf(&buf, "-%s", v.pre)
}
if v.metadata != "" {
 fmt.Fprintf(&buf, "+%s", v.metadata)
}

return buf.String()
}

// Original returns the original parsed version as-is, including any
// potential whitespace, `v` prefix, etc.
func (v *Version) Original() string {
 return v.original
}

// UnmarshalText implements encoding.TextUnmarshaler interface.
func (v *Version) UnmarshalText(b []byte) error {
 temp, err := NewVersion(string(b))
 if err != nil {
 return err
 }

 *v = *temp

 return nil
}

// MarshalText implements encoding.TextMarshaler interface.
func (v *Version) MarshalText() ([]byte, error) {
 return []byte(v.String()), nil
}

// Scan implements the sql.Scanner interface.
func (v *Version) Scan(src interface{}) error {
 switch src := src.(type) {
 case string:
 return v.UnmarshalText([]byte(src))
 case nil:
 return nil
 }
}

```

```

default:
 return fmt.Errorf("cannot scan %T as Version", src)
}
}

//
Value implements the driver.Valuer interface.
func (v *Version) Value() (driver.Value, error) {
 return v.String(), nil
}

```

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LGPL-2.1 OR BSD-2-Clause

## 1.43 iputils-clockdiff 20240905-r0

### 1.43.1 Available under license :

arping: GPL-2.0-or-later  
clockdiff: BSD-3-Clause  
ping: BSD-3-Clause  
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another



language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work  
are not derived from the Program,  
and can be reasonably considered independent and separate works in  
themselves, then this License, and its terms, do not apply to those  
sections when you distribute them as separate works. But when you  
distribute the same sections as part of a whole which is a work based  
on the Program, the distribution of the whole must be on the terms of  
this License, whose permissions for other licensees extend to the  
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest  
your rights to work written entirely by you; rather, the intent is to  
exercise the right to control the distribution of derivative or  
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program  
with the Program (or with a work based on the Program) on a volume of  
a storage or distribution medium does not bring the other work under  
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable  
source code, which must be distributed under the terms of Sections  
1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three  
years, to give any third party, for a charge no more than your  
cost of physically performing source distribution, a complete  
machine-readable copy of the corresponding source code, to be  
distributed under the terms of Sections 1 and 2 above on a medium  
customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer  
to distribute corresponding source code. (This alternative is  
allowed only for noncommercial distribution  
and only if you  
received the program in object code or executable form with such  
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for  
making modifications to it. For an executable work, complete source  
code means all the source code for all modules it contains, plus any  
associated interface definition files, plus the scripts used to  
control compilation and installation of the executable. However, as a  
special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by

all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than ``show w'` and ``show c'`; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this  
is what you want to do, use the GNU Lesser General  
Public License instead of this License.

# 1.44 iputils-tracepath 20240905-r0

## 1.44.1 Available under license :

arping: GPL-2.0-or-later  
clockdiff: BSD-3-Clause  
ping: BSD-3-Clause  
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software

Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION



0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide

a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions

either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least

the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU

General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may

consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.45 iputils-ping 20240905-r0

### 1.45.1 Available under license :

arping: GPL-2.0-or-later  
clockdiff: BSD-3-Clause  
ping: BSD-3-Clause  
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and



modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer

to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to

this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.46 iputils-arping 20240905-r0

### 1.46.1 Available under license :

arping: GPL-2.0-or-later

clockdiff: BSD-3-Clause

ping: BSD-3-Clause

tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free



program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause

any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the

Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if

necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this  
is what you want to do, use the GNU Lesser General  
Public License instead of this License.

## 1.47 alpine-baselayout 3.6.8-r1

### 1.47.1 Available under license :

Note that many files in here are not copyrighted/licensed, so they are  
actually in the public domain.

Otherwise:

Copyright 1996-2013 Gentoo Foundation

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This  
General Public License applies to most of the Free Software  
Foundation's software and to any other program whose authors commit to  
using it. (Some other Free Software Foundation software is covered by  
the GNU Library General Public License  
instead.) You can apply it to  
your programs, too.

When we speak of free software, we are referring to freedom, not  
price. Our General Public Licenses are designed to make sure that you

have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another



language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place,  
then offering equivalent  
access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program),  
you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License

and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by

all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you

can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type  
'show w'.

This is free software, and you are welcome to redistribute it  
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than 'show w' and 'show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit  
linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

# 1.48 musl 1.2.5-r9

## 1.48.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

-----  
Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----

Authors/contributors include:

A. Wilcox  
Ada Worcester  
Alex Dowad  
Alex Suykov  
Alexander Monakov  
Andre McCurdy  
Andrew Kelley  
Anthony G. Basile  
Aric Belsito  
Arvid Picciani  
Bartosz Brachaczek  
Benjamin Peterson  
Bobby Bingham  
Boris Brezillon  
Brent Cook

Chris Spiegel  
Clment Vasseur  
Daniel Micay  
Daniel Sabogal  
Daurnimator  
David Carlier  
David Edelsohn  
Denys Vlasenko  
Dmitry Ivanov  
Dmitry V. Levin  
Drew DeVault  
Emil Renner Berthing  
Fangrui Song  
Felix Fietkau  
Felix Janda  
Gianluca Anzolin  
Hauke Mehrrens  
He X  
Hiltjo Posthuma  
Isaac Dunham  
Jaydeep Patil  
Jens Gustedt  
Jeremy Huntwork  
Jo-Philipp Wich  
Joakim Sindholt  
John Spencer  
Julien Ramseier  
Justin Cormack  
Kaarle Ritvanen  
Khem Raj  
Kylie McClain  
Leah Neukirchen  
Luca Barbato  
Luka  
Perkov  
M Farkas-Dyck (Strake)  
Mahesh Bodapati  
Markus Wichmann  
Masanori Ogino  
Michael Clark  
Michael Forney  
Mikhail Kremnyov  
Natanael Copa  
Nicholas J. Kain  
orc  
Pascal Cuoq  
Patrick Oppenlander  
Petr Hosek



Petr Skocik  
Pierre Carrier  
Reini Urban  
Rich Felker  
Richard Pennington  
Ryan Fairfax  
Samuel Holland  
Segev Finer  
Shiz  
sin  
Solar Designer  
Stefan Kristiansson  
Stefan O'Rear  
Szabolcs Nagy  
Timo Ters  
Trutz Behn  
Valentin Ochs  
Will Dietz  
William Haddon  
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (`src/string/aarch64/*`) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (src/crypt/crypt\_des.c) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (src/crypt/crypt\_blowfish.c) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (src/stdlib/qsart.c) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86\_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the ellcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/\* and arch/\*/bits/\*) and crt files intended to be linked into applications (crt/\*, ldso/dlstart.c, and arch/\*/crt\_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham  
John Spencer  
Nicholas J. Kain  
Rich Felker  
Richard Pennington

Stefan Kristiansson  
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated

the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

## 1.49 stdr 1.2.2

### 1.49.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.50 go-glob 1.0.0

## 1.50.1 Available under license :

MIT License

Copyright (c) 2019 Donald King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.51 go-secure-stdlib-strutil 0.1.2

### 1.51.1 Available under license :

Mozilla Public License, version 2.0

#### 1. Definitions

##### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

##### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

##### 1.3. "Contribution"

means Covered Software of a particular Contributor.

##### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

##### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or



b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

#### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

#### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. "License"

means this document.

#### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications

that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support,

indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

#### 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

#### 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject

matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible

With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible  
With Secondary Licenses", as defined by  
the Mozilla Public License, v. 2.0.

## 1.52 ca-certificates 20241121-r1

### 1.52.1 Available under license :

```
/* c_rehash.c - Create hash symlinks for certificates
 * C implementation based on the original Perl and shell versions
 *
 * Copyright (c) 2013-2014 Timo Teräs <timo.teras@iki.fi>
 * All rights reserved.
 *
 * This software is licensed under the MIT License.
 * Full license available at: http://opensource.org/licenses/MIT
 */
Contributor: Sören Tempel <soeren+alpine@soeren-tempel.net>
Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=ca-certificates
pkgver=20241121
pkgrel=1
pkgdesc="Common CA certificates PEM files from Mozilla"
url="https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/"
arch="all"
There is a GPL-2.0-or-later script inside the source but it is not shipped
license="MPL-2.0 AND MIT"
makedepends_build="perl"
Mozilla Public License
Version 2.0
1. Definitions
1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
```



means Covered Software of a particular Contributor.

1.4. “Covered Software”

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. “Incompatible With Secondary Licenses”

means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form”

means

any form of the work other than Source Code Form.

1.7. “Larger Work”

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License”

means this document.

1.9. “Licensable”

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications”

means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License”

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. “Source Code Form”

means the form of the work preferred for making modifications.

#### 1.14. “You” (or “Your”)

means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for

any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

### 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

### 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has

sufficient rights  
to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in

Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s),

so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred

by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

#### 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

#### 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

#### 6. Disclaimer of Warranty

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

#### 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent

applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

# 1.53 busybox 1.37.0-r12

## 1.53.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source  
of bzip2 version 1.0.4. bzip2 source is distributed  
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all  
documentation, are copyright (C) 1996-2006 Julian R Seward. All  
rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must  
not claim that you wrote the original software. If you use this  
software in a product, an acknowledgment in the product  
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must  
not be misrepresented  
as being the original software.
4. The name of the author may not be used to endorse or promote  
products derived from this software without specific prior written  
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS  
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY  
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE  
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,  
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING  
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the



notices that refer to this License and to the absence of any warranty;  
and give any other recipients of the Program a copy of this License  
along with the Program.

You may charge a fee for the physical act of transferring a copy, and  
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion  
of it, thus forming a work based on the Program, and  
copy and  
distribute such modifications or work under the terms of Section 1  
above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices  
stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in  
whole or in part contains or is derived from the Program or any  
part thereof, to be licensed as a whole at no charge to all third  
parties under the terms of this License.
- c) If the modified program normally reads commands interactively  
when run, you must cause it, when started running for such  
interactive use in the most ordinary way, to print or display an  
announcement including an appropriate copyright notice and a  
notice that there is no warranty (or else, saying that you provide  
a warranty) and that users may redistribute the program under  
these conditions, and telling the user how to view a copy of this  
License. (Exception: if the Program itself is interactive but  
does not normally print such an announcement, your work based on  
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If  
identifiable sections of that work are not derived from the Program,  
and can be reasonably considered independent and separate works in  
themselves, then this License, and its terms, do not apply to those  
sections when you distribute them as separate works. But when you  
distribute the same sections as part of a whole which is a work based  
on the Program, the distribution of the whole must be on the terms of  
this License, whose permissions for other licensees extend to the  
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest  
your rights to work written entirely by you; rather, the intent is to  
exercise the right to control the distribution  
of derivative or  
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.54 ssl-client 1.37.0-r12

### 1.54.1 Available under license :

```
Automatically generated by apkgbuild-cpan, template 4
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
```

```

depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/${_pkgreal-$pkgver}"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
 INSTALLDIRS=vendor \
 NO_PACKLIST=1 \
 NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
 BSD-Resource-1.2911.tar.gz
"

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)
=====

```

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this

software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This



General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you

received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# Contributor: Maxim Karasev <begs@disroot.org>

# Maintainer: mio <miyopan@e.email>

pkgname=bsd-games

pkgver=3.3

pkgrel=2

pkgdesc="traditional text mode games from BSD"

url="https://bsd-games.sourceforge.io/"

arch="all"

license="BSD-3-Clause"

install="\$pkgname.post-install"

makedepends="ncurses-dev coreutils" # configure causes busybox expr error

subpackages="\$pkgname-doc"

source="https://sourceforge.net/projects/bsd-games/files/bsd-games-\$pkgver.tar.gz

0001-bsd-games-scores-dir.patch

"

options="!check" # no tests

build() {

# some GNU autoconf options are ignored, but it works

./configure \

--build=\$CBUILD \

--host=\$CHOST \

--prefix=/usr \

--sysconfdir=/etc \

--mandir=/usr/share/man \

--sourcedir=/usr/share/\$pkgname

make

}

package() {

make DESTDIR="\$pkgdir" install

install -Dm644 LICENSE "\$pkgdir"/usr/share/licenses/bsd-games/LICENSE

}

sha512sums="

aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6  
8d3390143e0e62d5e3d210ccfb61294

bsd-games-3.3.tar.gz

e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0  
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch

"



MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Vintage Dreams Waves v 2.0. for Creative Labs' AWE Soundcards  
(EMU Soundfont 2 Format)

Copyright (c) Ian Wilson, 1996 (Updated January 1998)

This soundfont is freeware. You may freely use and/or redistribute it subject to the following terms:

1. It is not altered, edited, modified, ripped, or converted to other formats, except for private use only.
2. It is distributed with this copyright notice.

This soundfont is distributed WITHOUT WARRANTY, and without the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. No liability or damages can be inferred upon the said copyright owner, Ian Wilson.

Any feedback, contact Ian Wilson.

[vintagedreamworks@hotmail.com](mailto:vintagedreamworks@hotmail.com)

<http://www.geocities.com/SiliconValley/Campus/8645/index.html>

<http://members.nbci.com/silicon39/>

<http://www.mp3.com/silicon39>

<https://analoguesque.x10host.com/>

# Maintainer: Natanael Copa <[ncopa@alpinelinux.org](mailto:ncopa@alpinelinux.org)>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

```
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"
```

```
builddir="$srcdir"/libart_lgpl-$pkgver
```

```
prepare() {
 update_config_sub
 default_prepare
}
```

```
build() {
 ./configure \
 --build=$CBUILD \
 --host=$CHOST \
 --prefix=/usr
 make
}
```

```
package() {
 make DESTDIR="$pkgdir" install
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING,  
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

subpackages="\$pkgname-list"

\_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in \$\_types; do

subpackages="\$subpackages \$pkgname-\$type:\_subpkg"

done

package() {

mkdir -p "\$pkgdir"

}

\_subpkg() {

local type=\${subpkgname/\$pkgname-/}

pkgdesc="\$pkgdesc (\$type)"

install\_if="\$pkgname=\$pkgver-r\$pkgrel"

mkdir -p "\$subpkgdir"/usr/share/spdx

cp -r "\$builddir"/\$type "\$subpkgdir"/usr/share/spdx/

}

list() {

pkgdesc="\$pkgdesc (licence list)"

mkdir -p "\$subpkgdir"/usr/share/spdx

local i; for i in "\$builddir"/text/\*.txt;

do

local license=\${i##\*/}

echo \${license%.\*} >> "\$subpkgdir"/usr/share/spdx/license.lst

done

}

```

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

Contributor: Patrycja Rosa <alpine@ptrcnll.me>
Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

buildddir="$srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
 install -Dm644 -t "$pkgdir"/usr/include/sys \
 cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

```

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w\_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

# Contributor: Dhruvin Gandhi <contact@dhruvin>

# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

\_pkgname=license\_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools\_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\_pkgname/\_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\_pkgname-\$pkgver"

build() {

gpep517 build-wheel \

--wheel-dir .dist \

--output-fd 3 3>&1 >&2

}

check() {

python3 -m venv --clear --without-pip --system-site-packages .testenv

.testenv/bin/python3 -m installer .dist/\*.whl

.testenv/bin/python3 -m pytest

}

package() {

python3 -m installer -d "\$pkgdir" \

.dist/\*.whl

```

}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
 license_expression-30.4.1.tar.gz
"

Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
 gpep517 build-wheel \
 --wheel-dir .dist \
 --output-fd 3 3>&1 >&2
}

package() {
 gpep517 install-wheel --destdir "$pkgdir" \
 .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
 flake8-copyright-0.2.4.tar.gz
"

Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"

```

```

url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}

build() {
 cargo auditable build --release --frozen
}

check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

```

## 1.55 s2a-go 0.1.9

### 1.55.1 Available under license :

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."



"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.56 ca-certificates-bundle 20241121-r1

### 1.56.1 Available under license :

```
Contributor: Patrycja Rosa <alpine@ptrcnull.me>
Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
buildid=" $srcdir"
options="!check" # just headers

package() {
 mkdir -p "$pkgdir"
 install -Dm644 -t "$pkgdir"/usr/include/sys \
 cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
```

```

source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2
"

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
 update_config_sub
 default_prepare
}

build() {
 ./configure \
 --build=$CBUILD \
 --host=$CHOST \
 --prefix=/usr
 make
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"

```

## 1.57 alpine-baselayout-data 3.6.8-r1

### 1.57.1 Available under license :

```

Automatically generated by apkbuild-cpan, template 1
Contributor: Francesco Colista <francesco.colista@gmail.com>
Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=""
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/_pkgreal-$pkgver.tar.gz"

_buildddir="$srcdir/_pkgreal-$pkgver"

```

```
prepare() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}
```

```
build() {
 cd "$_builddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test
}
```

```
package() {
 cd "$_builddir"
 make
 DESTDIR="$pkgdir" install || return 1
 find "$pkgdir" \(-name perllocal.pod -o -name .packlist \) -delete
}
```

md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"

**IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.**

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

## 1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

## 2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

### 3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition, Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered

Code without prior written authorization from Intel.

#### 4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

/\*

\* Copyright (c) 1983, 1988 Regents of the University of California.

\* All rights reserved.

\*



- \* Redistribution and use in source and binary forms are permitted
- \* provided that the above copyright notice and this paragraph are
- \* duplicated in all such forms and that any documentation,
- \* advertising materials, and other materials related to such
- \* distribution and use acknowledge that the software was developed
- \* by the University of California, Berkeley. The name of the
- \* University may not be used to endorse or promote products derived
- \* from this software without specific prior written permission.
- \* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
- \* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
- \* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
- \*/

## INTEL SOFTWARE LICENSE AGREEMENT

### IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

### LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

**SITE LICENSE.** You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

**SINGLE USER LICENSE.** You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights,

trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

**LIMITED MEDIA WARRANTY.** If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

**EXCLUSION OF OTHER WARRANTIES.** EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

**LIMITATION OF LIABILITY.** IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

**TERMINATION OF THIS AGREEMENT.** Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

**APPLICABLE LAWS.** Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

**GOVERNMENT RESTRICTED RIGHTS.** The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel  
2200 Mission College Blvd., Santa Clara, CA  
95052.

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>

# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

```

pkgname=py-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.0
pkgrel=0
pkgdesc="Extension for flake8 which checks for copyrights"
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="flake8"
makedepends="python3-dev"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/${_pkgname}-${pkgver}

build() {
 cd "$builddir"
 python3 setup.py build || return 1
}

package() {
 cd "$builddir"
 python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1
}

md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-0.2.0.tar.gz"
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e2417b586e543bfba12abce256b2023c8c4993eeb90flake8-copyright-0.2.0.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have

public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

#### Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.

Your legal department tells

you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

#### Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:

The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci  
6200 Maple Cove Lane  
Charlotte, NC 28269  
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>

Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is

distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation, details at:  
<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.  
mini\_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=5

pkgdesc="A library for high-performance 2D graphics"

url="http://www.levien.com/libart/"

arch="all"

license="LGPL"

depends=

makedepends=

subpackages="\$pkgname-dev"

source="http://ftp.gnome.org/pub/GNOME/sources/libart\_lgpl/2.3/libart\_lgpl-\$pkgver.tar.bz2"

"

\_builddir="\$srcdir"/libart\_lgpl-\$pkgver

prepare() {

cd "\$\_builddir"

update\_config\_sub || return 1

}

build() {

cd "\$\_builddir"

./configure \

--build=\$CBUILD \

--host=\$CHOST \

--prefix=/usr \

|| return 1

make || return 1

}

package() {

cd "\$\_builddir"

make DESTDIR="\$pkgdir" install || return 1

}

md5sums="08559ff3c67fd95d57b0c5e91a6b4302 libart\_lgpl-2.3.21.tar.bz2"

# Automatically generated by apkbuild-cpan, template 1

# Contributor: Valery Kartel <valery.kartel@gmail.com>

# Maintainer: Valery Kartel <valery.kartel@gmail.com>

pkgname=perl-bsd-resource

\_pkgreal=BSD-Resource

pkgver=1.2909

pkgrel=1

pkgdesc="Perl extension implements the BSD process resource limit functions"

url="http://search.cpan.org/dist/BSD-Resource/"

arch="all"

```

license="GPL PerlArtistic"
cpandepends=""
cpanmakedepends=""
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/${_pkgreal}-${pkgver}.tar.gz"

_buildddir="${srcdir}/${_pkgreal}-${pkgver}"

prepare() {
 cd "$_buildddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
 cd "$_buildddir"
 export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
 make && make test || return 1
}

package() {
 cd "$_buildddir"
 make DESTDIR="$pkgdir" install || return 1
 find "$pkgdir"
 \(-name perllocal.pod -o -name .packlist\) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"

```

## 1.58 github.com/google/s2a-go 0.1.9

### 1.58.1 Available under license :

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.



"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.59 go-cleanhttp 0.5.2

### 1.59.1 Available under license :

Mozilla Public License, version 2.0

#### 1. Definitions

##### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

##### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

##### 1.3. "Contribution"

means Covered Software of a particular Contributor.

##### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

##### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the

License, but not also under the terms of  
a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims,  
in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or

- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not



attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by

You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

#### 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

#### 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not

be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible

With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at

<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible  
With Secondary Licenses", as defined by  
the Mozilla Public License, v. 2.0.

## 1.60 busybox 1.37.0-r14

### 1.60.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source  
of bzip2 version 1.0.4. bzip2 source is distributed  
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all  
documentation, are copyright (C) 1996-2006 Julian R Seward. All  
rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must  
not claim that you wrote the original software. If you use this  
software in a product, an acknowledgment in the product  
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must  
not be misrepresented  
as being the original software.
4. The name of the author may not be used to endorse or promote  
products derived from this software without specific prior written  
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS  
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY  
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program

is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the



entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering

access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other

circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of
Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.61 openssl 3.3.3-r0

### 1.61.1 Available under license :

```
/*
 * Copyright 2006-2020 The OpenSSL Project Authors. All Rights Reserved.
 *
 * Licensed under the Apache License 2.0 (the "License"). You may not use
 * this file except in compliance with the License. You can obtain a copy
```

\* in the file LICENSE in the source distribution or at  
\* <https://www.openssl.org/source/license.html>  
\*/

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.



Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.62 libcrypto3 3.3.3-r0

## 1.62.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE  
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free

programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free

programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,  
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.



3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
- b) use the modified Package only within your corporation or organization.
- c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
- b) accompany the distribution with the machine-readable source of the Package with your modifications.
- c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright

of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.63 libssl3 3.3.3-r0

### 1.63.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE  
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The

General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding



machine-readable

source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these

terms.

To do so,  
attach the following notices to the program. It is safest to  
attach them to the start of each source file to most effectively convey  
the exclusion of warranty; and each file should have at least the  
"copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 1, or (at your option)  
any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free  
Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the  
appropriate parts of the General Public License. Of course, the  
commands you use may be called something other than `show w' and `show  
c'; they could even be mouse-clicks or menu items--whatever suits your  
program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here a sample; alter  
the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
  - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
  - b) use the modified Package only within your corporation or organization.
  - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
  - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
  - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
  - b) accompany the distribution with the machine-readable source of the Package with your modifications.
  - c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
  - d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.
7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.
8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or



documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,  
incidental, or consequential damages of any character arising as a  
result of this License or out of the use or inability to use the  
Work (including but not limited to damages for loss of goodwill,  
work stoppage, computer failure or malfunction, or any and all  
other commercial damages or losses), even if such Contributor  
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing  
the Work or Derivative Works thereof, You may choose to offer,  
and charge a fee for, acceptance of support, warranty, indemnity,  
or other liability obligations and/or rights consistent with this  
License. However, in accepting such obligations, You may act only  
on Your own behalf and on Your sole responsibility, not on behalf  
of any other Contributor, and only if You agree to indemnify,  
defend, and hold each Contributor harmless for any liability  
incurred by, or claims asserted against, such Contributor by reason  
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.64 alpine-base 3.21.3-r0

### 1.64.1 Available under license :

The MIT License

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Contributor: Natanael Copa <ncopa@alpinelinux.org>

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=alpine-base

pkgver=3.21.3

pkgrel=0

```
pkgdesc="Meta package for minimal alpine base"
url="https://alpinelinux.org"
arch="noarch"
license="MIT"
depends="
```

## 1.65 musl-utils 1.2.5-r9

### 1.65.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

-----  
Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----

Authors/contributors include:

A. Wilcox  
Ada Worcester  
Alex Dowad  
Alex Suykov  
Alexander Monakov  
Andre McCurdy  
Andrew Kelley  
Anthony G. Basile  
Aric Belsito  
Arvid Picciani

Bartosz Brachaczek  
Benjamin Peterson  
Bobby Bingham  
Boris Brezillon  
Brent Cook  
Chris Spiegel  
Clment Vasseur  
Daniel Micay  
Daniel Sabogal  
Daurnimator  
David Carlier  
David Edelsohn  
Denys Vlasenko  
Dmitry Ivanov  
Dmitry V. Levin  
Drew DeVault  
Emil Renner Berthing  
Fangrui Song  
Felix Fietkau  
Felix Janda  
Gianluca Anzolin  
Hauke Mehrrens  
He X  
Hiltjo Posthuma  
Isaac Dunham  
Jaydeep Patil  
Jens Gustedt  
Jeremy Huntwork  
Jo-Philipp Wich  
Joakim Sindholt  
John Spencer  
Julien Ramseier  
Justin Cormack  
Kaarle Ritvanen  
Khem Raj  
Kylie McClain  
Leah Neukirchen  
Luca Barbato  
Luka  
Perkov  
M Farkas-Dyck (Strake)  
Mahesh Bodapati  
Markus Wichmann  
Masanori Ogino  
Michael Clark  
Michael Forney  
Mikhail Kremnyov  
Natanael Copa

Nicholas J. Kain  
orc  
Pascal Cuoq  
Patrick Oppenlander  
Petr Hosek  
Petr Skocik  
Pierre Carrier  
Reini Urban  
Rich Felker  
Richard Pennington  
Ryan Fairfax  
Samuel Holland  
Segev Finer  
Shiz  
sin  
Solar Designer  
Stefan Kristiansson  
Stefan O'Rear  
Szabolcs Nagy  
Timo Ters  
Trutz Behn  
Valentin Ochs  
Will Dietz  
William Haddon  
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD

license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (src/string/aarch64/\*) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (src/crypt/crypt\_des.c) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (src/crypt/crypt\_blowfish.c) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (src/stdlib/qsort.c) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86\_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the elcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/\* and arch/\*/bits/\*) and crt files intended to be linked into applications (crt/\*, ldso/dlstart.c, and arch/\*/crt\_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham  
John Spencer  
Nicholas J. Kain  
Rich Felker  
Richard Pennington  
Stefan Kristiansson  
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated

the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

## 1.66 apk-tools 2.14.6-r3

### 1.66.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you

wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program



is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend

to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with

the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software

distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING  
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR  
REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY  
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE  
OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA  
OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES  
OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH  
HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.  
END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible  
use to the public, the best  
way to achieve this is to make it free software  
which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach  
them to the start of each source file to most effectively convey the exclusion  
of warranty; and each file should have at least the "copyright" line and a  
pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under  
the terms of the GNU General Public License as published by the Free Software  
Foundation; either version 2 of the License, or (at your option) any later  
version.

This program is distributed in the hope that it will be useful, but WITHOUT  
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS  
FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a  
copy of the GNU General Public License along with  
this program; if not, write to the Free Software Foundation, Inc., 51 Franklin  
Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when  
it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes

with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice  
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

# 1.67 golang-genproto 0.0.0

## 1.67.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one



of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.68 go.opentelemetry.io/auto/sdk 1.1.0

## 1.68.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.69 github.com/go-logr/stdr 1.2.2

### 1.69.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,



worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,  
each Contributor hereby grants to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the  
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding  
those notices that do not  
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.70 alpine-release 3.21.3-r0

## 1.70.1 Available under license :

```
Automatically generated by apkbuild-cpan, template 4
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/${_pkgreal}-${pkgver}"

build() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
 INSTALLDIRS=vendor \
 NO_PACKLIST=1 \
 NO_PERLLOCAL=1
 make
}

check() {
 export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
 make test
}

package() {
 make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
 BSD-Resource-1.2911.tar.gz
"

Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
```

```

pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
 default_prepare

 cargo fetch --target="$CTARGET" --locked
}

build() {
 cargo auditable build --release --frozen
}

check() {
 cargo test --frozen
}

package() {
 install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"
MIT
MIT License

```

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Contributor: Dhruvin Gandhi <contact@dhruvin>

# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>

pkgname=py3-license-expression

\_pkgname=license\_expression

pkgver=30.4.0

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools\_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\${\_pkgname}/\${\_pkgname}-\${pkgver}.tar.gz"

builddir="\$srcdir/\${\_pkgname}-\${pkgver}"

build() {

gpep517 build-wheel \

--wheel-dir .dist \

--output-fd 3 3>&1 >&2

}

check() {

python3 -m venv --clear --without-pip --system-site-packages .testenv

.testenv/bin/python3 -m installer .dist/\*.whl

.testenv/bin/python3 -m pytest

}

package() {

python3 -m installer -d "\$pkgdir" \

.dist/\*.whl

}

sha512sums="

cc7cd6b5bd8ea4c99b66f074871ffd7cc78b3e8b1b0d3afd9b95efdb7147c8293db2fac977d684fc34d3d60bd79066a06  
dc2bff0a38b130d9d83c63538502db1

license\_expression-30.4.0.tar.gz

"

Vintage Dreams Waves v 2.0. for Creative Labs' AWE Soundcards  
(EMU Soundfont 2 Format)

Copyright (c) Ian Wilson, 1996 (Updated January 1998)

This soundfont is freeware. You may freely use and/or redistribute it subject to the following terms:

1. It is not altered, edited, modified, ripped, or converted to other formats, except for private use only.
2. It is distributed with this copyright notice.

This soundfont is distributed WITHOUT WARRANTY, and without the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. No liability or damages can be inferred upon the said copyright owner, Ian Wilson.

Any feedback, contact Ian Wilson.

[vintagedreamworks@hotmail.com](mailto:vintagedreamworks@hotmail.com)

<http://www.geocities.com/SiliconValley/Campus/8645/index.html>

<http://members.nbc.com/silicon39/>

<http://www.mp3.com/silicon39>

<https://analoguesque.x10host.com/>

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

mini\_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

arch="all"

options="!check" # No test suite.

license="LGPL-2.0-or-later"

subpackages="\$pkgname-dev"

source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-\$pkgver.tar.bz2"

"

builddir="\$srcdir"/libart-lgpl-\$pkgver

```
prepare() {
 update_config_sub
 default_prepare
}
```

```
build() {
 ./configure \
 --build=$CBUILD \
 --host=$CHOST \
 --prefix=/usr
 make
}
```

```
package() {
 make DESTDIR="$pkgdir" install
}
```



```
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
```

## 1.71 vault-api 1.16.0

### 1.71.1 Available under license :

```
{{!
```

```
Copyright (c) HashiCorp, Inc.
```

```
SPDX-License-Identifier: BUSL-1.1
```

```
~}}
```

```
<LicenseInfo
```

```
@startTime={{this.model.startTime}}
```

```
@expirationTime={{this.model.expirationTime}}
```

```
@licenseId={{this.model.licenseId}}
```

```
@features={{this.model.features}}
```

```
@autoloaded={{this.model.autoloaded}}
```

```
@performanceStandbyCount={{this.model.performanceStandbyCount}}
```

```
/>
```

```

```

```
layout: docs
```

```
page_title: Automated license utilization reporting
```

```
description: >-
```

```
Learn what data HashiCorp collects to meter Enterprise license utilization. Enable or disable reporting. Review
sample payloads and logs.
```

```

```

```
Automated license utilization reporting
```

```
@include 'alerts/enterprise-only.mdx'
```

Automated license utilization reporting sends license utilization data to HashiCorp without requiring you to manually collect and report them. It also lets you review your license usage with the monitoring solution you already use (for example Splunk, Datadog, or others) so you can optimize and manage your deployments. Use these reports to understand how much more you can deploy under your current contract, protect against overutilization, and budget for predicted consumption.

Automated reporting shares the minimum data required to validate license utilization as defined in our contracts. They consist of mostly computed metrics and will never contain Personal Identifiable Information

(PII) or other

sensitive information. Automated reporting shares the data with HashiCorp using a secure, unidirectional HTTPS API and makes an auditable record in the product

logs each time it submits a report. The reporting process is GDPR compliant and submits reports roughly once every 24 hours.

## ## Enable automated reporting

To enable automated reporting, you need to make sure that outbound network traffic is configured correctly and upgrade your enterprise product to a version that supports it. If your installation is air-gapped or network settings are not in place, automated reporting will not work.

### ### 1. allow outbound HTTPS traffic on port 443

Make sure that your network allows HTTPS egress on port 443 to `https://reporting.hashicorp.services` by allow-listing the following IP addresses:

- 100.20.70.12
- 35.166.5.222
- 23.95.85.111
- 44.215.244.1

### ### 2. upgrade

Upgrade to a release that supports license utilization reporting. These releases include:

- [Vault Enterprise 1.14.0](https://releases.hashicorp.com/vault/) and later
- [Vault Enterprise 1.13.4](https://releases.hashicorp.com/vault/) and later 1.13.x versions
- [Vault Enterprise 1.12.8](https://releases.hashicorp.com/vault/) and later 1.12.x versions
- [Vault Enterprise 1.11.12](https://releases.hashicorp.com/vault/)

### ### 3. check logs

Automatic license utilization reporting will start sending data within roughly 24 hours. Check the server logs for records that the data sent successfully.

You will find log entries similar to the following:

<CodeBlockConfig hideClipboard>

...

```
[DEBUG] core.reporting: beginning snapshot export
[DEBUG] core.reporting: creating payload
[DEBUG] core.reporting: marshalling payload to json
[DEBUG] core.reporting: generating authentication headers
[DEBUG] core.reporting: creating request
[DEBUG] core.reporting: sending request
```

```
[DEBUG] core.reporting: performing request: method=POST url=https://reporting.hashicorp.services
[DEBUG] core.reporting: recording audit
record
[INFO] core.reporting: Report sent:
auditRecord="{\"payload\":{\"payload_version\":\"1\",\"license_id\":\"97afe7b4-b9c8-bf19-bf35-
b89b5cc0feaf\",\"product\":\"vault\",\"product_version\":\"1.14.0-rc1+ent\",\"export_timestamp\":\"2023-06-
01T09:34:44.215133-
04:00\",\"snapshots\":{\"snapshot_version\":\"1\",\"snapshot_id\":\"0001J7H7KMEDRXKM5C1QJGBXV3\",\"proces
s_id\":\"01H1T45CZK2GN9WR22863W2K32\",\"timestamp\":\"2023-06-01T09:34:44.215001-
04:00\",\"schema_version\":\"1.0.0\",\"service\":\"vault\",\"metrics\":{\"clientcount.current_month_estimate\":{\"key
\":\"clientcount.current_month_estimate\",\"kind\":\"sum\",\"mode\":\"write\",\"labels\":{\"type\":{\"entity\":\"20\",\"no
nentity\":\"11\"}},\"clientcount.previous_month_complete\":{\"key\":\"clientcount.previous_month_complete\",\"kind
\":\"sum\",\"mode\":\"write\",\"labels\":{\"type\":{\"entity\":\"10\",\"nonentity\":\"11\"}}}},\"metadata\":{\"vault\":{\"billi
ng_start\":\"2023-03-01T00:00:00Z\",\"cluster_id\":\"a8d95acc-ec0a-6087-d7f6-4f054ab2e7fd\"}}}"
[DEBUG]
core.reporting: completed recording audit record
[DEBUG] core.reporting: export finished successfully
...
```

</CodeBlockConfig>

If your installation is air-gapped or your network doesn't allow the correct egress, logs will show an error.

<CodeBlockConfig hideClipboard>

```
...
[DEBUG] core.reporting: beginning snapshot export
[DEBUG] core.reporting: creating payload
[DEBUG] core.reporting: marshalling payload to json
[DEBUG] core.reporting: generating authentication headers
[DEBUG] core.reporting: creating request
[DEBUG] core.reporting: sending request
[DEBUG] core.reporting: performing request: method=POST url=https://reporting.hashicorp.services
[DEBUG] core.reporting: error status code received: statusCode=403
...
```

</CodeBlockConfig>

In this case, reconfigure your network to allow egress and check back in 24 hours.

## Opt out

If your installation is air-gapped or you want to manually collect and report on the same license utilization metrics, you can opt-out of automated reporting.

Manually reporting these metrics can be time consuming. Opting out of automated reporting does not mean that you also opt out from sending license utilization metrics. Customers who opt out of automated reporting will still be required to manually collect and send license utilization metrics to HashiCorp.

If you are considering opting out because you're worried about the data, we strongly recommend that you review the [example payloads](#example-payloads) before opting out. If you have concerns with any of the automatically-reported data please bring them to your account manager.

You have two options to opt out of automated reporting:

- HCL configuration (recommended)
- Environment variable (requires restart)

#### #### HCL configuration

Opting out in your products configuration file doesn't require a system restart, and is the method we recommend. Add the following block to your server configuration file (e.g. `vault-config.hcl`).

```
```hcl
reporting {
  license
  {
    enabled = false
  }
}
```
```

<Warning>

When you have a cluster, each node must have the reporting stanza in its configuration to be consistent. In the event of leadership change, nodes will use its server configuration to determine whether or not to opt-out the automated reporting. Inconsistent configuration between nodes will change the reporting status upon active unseal.

</Warning>

You will find the following entries in the server log.

<CodeBlockConfig hideClipboard>

```
```
```

[DEBUG] core: reloading automated reporting

```
[INFO] core: opting out of automated reporting
[DEBUG] activity: there is no reporting agent configured, skipping counts reporting
...
```

</CodeBlockConfig>

Environment variable

If you need to, you can also opt out using an environment variable, which will provide a startup message confirming that you have disabled automated reporting. This option requires a system restart.

<Note>

If the reporting stanza exists in the configuration file, the ``OPTOUT_LICENSE_REPORTING`` value overrides the configuration.

</Note>

Set the following environment variable.

```
```shell-session
$ export OPTOUT_LICENSE_REPORTING=true
```
```

Now, restart your [Vault servers](/vault/docs/commands/server) from the shell where you set the environment variable.

You will find the following entries in the server log.

<CodeBlockConfig hideClipboard>

...

```
[INFO] core: automated reporting disabled via environment variable: env=OPTOUT_LICENSE_REPORTING
[INFO] core: opting out of automated reporting
[DEBUG] activity: there is no reporting agent configured, skipping counts reporting
...
```

</CodeBlockConfig>

Check your product logs roughly 24 hours after opting out to make sure that the system isnt trying to send reports.

If your configuration file and environment variable differ, the environment variable setting will take precedence.

Example payloads

HashiCorp collects the following utilization data as JSON payloads:

- `payload\_version` - The version of this payload schema
- `license\_id` - The license ID for this product
- `product` - The product that this contribution is for
- `product\_version` - The product version this contribution is for
- `export\_timestamp` - The date and time for this contribution
- `snapshots` - An array of snapshot details. A snapshot is a structure that represents a single data collection
 - `snapshot\_version` - The version of the snapshot package that produced this snapshot
 - `snapshot\_id` - A unique identifier for this particular snapshot
 - `process\_id` - An identifier for the system that produced this snapshot
 - `timestamp` - The date and time for this snapshot
 - `schema\_version` - The version of the schema associated with this snapshot
 - `service` - The service that produced this snapshot (likely to be product name)
 - `metrics` - A map of representations of snapshot metrics contained within this snapshot
 - `key` - The key name associated with this metric
 - `kind`
 - The kind of metric (feature, counter, sum, or mean)
 - `mode` - The mode of operation associated with this metric (write or collect)
 - `labels` - The labels associated with each collected metric
 - `entity` - The sum of tokens generated for a unique client identifier
 - `nonentity` - The sum of tokens without an entity attached
- `metadata` - Optional product-specific metadata
 - `billing\_start` - The billing start date associated with the reporting cluster (license start date if not configured)
 - `cluster\_id` - The cluster UUID as shown by `vault status` on the reporting cluster

<CodeBlockConfig hideClipboard>

```
```json
{
 "payload_version": "1",
 "license_id": "97afe7b4-b9c8-bf19-bf35-b89b5cc0efea",
 "product": "vault",
 "product_version": "1.14.0-rc1+ent",
 "export_timestamp": "2023-06-01T11:39:00.76643-04:00",
 "snapshots": [
 {
 "snapshot_version": 1,
 "snapshot_id": "0001J7HEWM1PEHPMF5YZT8EV65",
```



Starting in Vault 1.9, the activity log records and de-duplicates non-entity tokens by using the namespace and token's policies to generate a unique identifier. Because Vault did not create identifiers for these tokens before 1.9, the activity log cannot know whether this token has been seen pre-1.9. To prevent inaccurate and inflated counts, the activity log will ignore any counts of non-entity tokens that were created before the upgrade and only the non-entity tokens from versions 1.9 and later will be counted.

See the client count [overview](/vault/docs/concepts/client-count) and [FAQ](/vault/docs/concepts/client-count/faq) for more information.

```
{{!
```

```
Copyright (c) HashiCorp, Inc.
```

```
SPDX-License-Identifier: BUSL-1.1
```

```
~}}
```

```
{{#if (and this.licenseExpired (not this.expiredDismissed))}}
```

```
<Hds::Alert
```

```
 @type="inline"
```

```
 @color="critical"
```

```
 @onDismiss={{fn this.dismissBanner "expired"}}
```

```
 data-test-license-banner-expired
```

```
 as |A|
```

```
>
```

```
 <A.Title>License expired</A.Title>
```

```
 <A.Description>
```

```
 Your Vault license expired on
```

```
 {{date-format @expiry "MMM d, yyyy"}}. Add a new license to your configuration and restart Vault.
```

```
 </A.Description>
```

```
 <A.Description class="has-top-margin-xs">
```

```
 <DocLink @path="/vault/tutorials/enterprise/hashicorp-enterprise-license">
```

```
 Read documentation
```

```
 <Icon @name="learn-link" />
```

```
 </DocLink>
```

```
 </A.Description>
```

```
</Hds::Alert>
```

```
{{else if (and (lte this.licenseExpiringInDays 30) (not this.warningDismissed))}}
```

```
<Hds::Alert
```

```
 @type="inline"
```

```
 @color="warning"
```

```
 @onDismiss={{fn this.dismissBanner "warning"}}
```

```
 data-test-license-banner-warning
```

```
 as |A|
```

```
>
```

```
 <A.Title>Vault license expiring</A.Title>
```

```
 <A.Description>
```

```
 Your Vault license will expire in
```

```
 {{this.licenseExpiringInDays}}
```

```
 days at
```

```
 {{date-format @expiry "hh:mm:ss a"}}
```

```
 on
```



```

 {{date-format @expiry "MMM d, yyyy"}}.
 {{if
 @autoloaded
 "Add a new license to your configuration."
 "Keep in mind that your next license will need to be autoloaded."
 }}
 </A.Description>
 <A.Description class="has-top-margin-xs">
 <DocLink @path="/vault/tutorials/enterprise/hashicorp-enterprise-license">
 Read documentation
 <Icon @name="learn-link" />
 </DocLink>
 </A.Description>
</Hds::Alert>
{{/if}}

```

---

layout: docs

page\_title: license get - Command

description: |-

The "license get" command reports on the license in use.

---

# license get

The `license get` command reports on the license in use. With the `-signed` flag, it will return the actual license string, but only if the license is stored, as opposed to [autoloaded](/vault/docs/enterprise/license/autoloading).

## Examples

Describe current license:

```
```shell-session
```

```
$ vault license get
```

```
Key          Value
```

```
---
```

```
expiration_time      2022-05-17T23:59:59Z
```

```
features             [HSM Performance Replication DR Replication MFA Sentinel Seal Wrapping Control Groups
Performance Standby Namespaces KMIP Entropy Augmentation Transform Secrets Engine Lease Count Quotas
Key Management Secrets Engine Automated Snapshots]
```

```
license_id           060d7820-fa59-f95c-832b-395db0aeb9ba
```

```
performance_standby_count 9999
```

```
start_time           2021-05-17T00:00:00Z
```

```
termination_time
```

```
2031-05-17T23:59:59Z
```

```
```
```

Get stored license string:

```
```shell-session
$ vault license get -signed
02MV4UU43BK5H...
```
```

Try to get stored license string when using autoloading:

```
```shell-session
$ vault license get -signed
License not found or using a temporary license.
```
```

## ## Usage

The following flags are available in addition to the [standard set of flags](/vault/docs/commands) included on all commands.

### ### Command options

- `-signed` (bool: false)`` - Return the signed license string when using a stored license.

Copyright (c) 2015 HashiCorp, Inc.

Mozilla Public License, version 2.0

## 1. Definitions

### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

### 1.3. "Contribution"

means Covered Software of a particular Contributor.

### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

### 2. License Grants and Conditions

#### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

#### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

#### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under

this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4.

### Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor

explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under

## Section

2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.



## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward.

Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can

obtain one at  
<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible  
With Secondary Licenses", as defined by  
the Mozilla Public License, v. 2.0.

---

layout: api

page\_title: /sys/license/status - HTTP API

description: |-

The `/sys/license/status`` endpoint is used to view the license used in Vault.

---

# `/sys/license/status``

@include 'alerts/enterprise-only.mdx'

The `/sys/license/status`` endpoint is used to view update the license used in Vault.

#### ## License status

This endpoint returns information about licensing. See [license autoloading](/vault/docs/enterprise/license/autoloading) for additional background.

In the response:

- ``autoloading_used`` will be true if an autoloaded license was provided to the node, false otherwise. This should always be true from Vault 1.11.0 onward.
- ``autoloaded`` is the autoloaded license if `autoloading_used` is true.
- ``persisted_autoload`` is the autoloaded license the active node last wrote to storage; this is only used to detect inconsistently licensed nodes in the cluster. It cannot be used as a license itself.
- ``stored`` was used for the stored license prior to Vault 1.11.0. Stored licenses are no longer supported.

``autoloaded``, and ``persisted_autoload`` will both have the same structure.

```
Method	Path
`GET`	`/sys/license/status`
```

### Sample request

```
```shell-session
$ curl \
  --header "X-Vault-Token: ..." \
  http://127.0.0.1:8200/v1/sys/license/status
```
```

### Sample response

```
```json
{
  "data": {
    "autoloading_used": true,
    "autoloaded": {
      "expiration_time": "2022-05-17T23:59:59Z",
      "features": [
        "HSM",
        "Performance Replication",
        "DR Replication",
        "MFA",
        "Sentinel",
        "Seal Wrapping",
        "Control Groups",
        "Performance Standby",
        "Namespaces",
        "KMIP",
        "Entropy Augmentation",
        "Transform Secrets Engine",
        "Lease Count Quotas",
        "Key Management Secrets Engine",
        "Automated Snapshots"
      ],
      "license_id": "060d7820-fa59-f95c-832b-395db0aeb9ba",
      "performance_standby_count":
        9999,
      "start_time": "2021-05-17T00:00:00Z",
      "termination_time": "2031-05-17T23:59:59Z"
    },
    "persisted_autoload": {
      "expiration_time": "2022-05-17T23:59:59Z",
      "features": [
        "HSM",
        "Performance Replication",

```

```

    "DR Replication",
    "MFA",
    "Sentinel",
    "Seal Wrapping",
    "Control Groups",
    "Performance Standby",
    "Namespaces",
    "KMIP",
    "Entropy Augmentation",
    "Transform Secrets Engine",
    "Lease Count Quotas",
    "Key Management Secrets Engine",
    "Automated Snapshots"
  ],
  "license_id": "060d7820-fa59-f95c-832b-395db0aeb9ba",
  "performance_standby_count": 9999,
  "start_time": "2021-05-17T00:00:00Z",
  "termination_time": "2031-05-17T23:59:59Z"
}
}
}
...
{{!
Copyright (c) HashiCorp, Inc.
SPDX-License-Identifier: BUSL-1.1
~}}

```

```

<PageHeader as |p|>
  <p.levelLeft>
    <h1 class="title is-3">License</h1>
  </p.levelLeft>
</PageHeader>

```

```

<section class="box is-sideless is-marginless is-shadowless is-fullwidth">
  <span class="title is-5">Details</span>
  <div class="field box is-fullwidth is-shadowless is-paddingless is-marginless">
    <InfoTableRow @label="License ID" @value={{@licenseId}} data-test-detail-row />
    <InfoTableRow @label="Valid from" @value={{@startTime}} data-test-detail-row>
      {{date-format @startTime "MMM dd, yyyy hh:mm:ss a"}}
    to
      {{date-format @expirationTime "MMM dd, yyyy hh:mm:ss a"}}
    </InfoTableRow>
    <InfoTableRow @label="License state" @value={{ "Autoloaded" }} data-test-detail-row />
  </div>
</section>

```

```

<section class="box is-sideless is-marginless is-shadowless is-fullwidth">
  <span class="title is-5">Features</span>

```

```

<div class="field box is-fullwidth is-shadowless is-paddingless is-marginless">
  {{#each this.featuresInfo as |info|}}
    <InfoTableRow @label={{info.name}} @value={{if info.active "Active" "Not Active"}} data-test-feature-row>
      {{#if info.active}}
        <Icon @name="check-circle" class="icon-true" />
        <span data-test-feature-status>
          Active
          {{#if info.count}}&mdash; {{info.count}} standby nodes allotted{{/if}}
        </span>
      {{else}}
        <Icon @name="x-circle" class="icon-false" />
        <span data-test-feature-status>Not Active</span>
      {{/if}}
    </InfoTableRow>
  {{/each}}
</div>
</section>

```

The MIT License (MIT)

Copyright (c) 2015 Andrew Smith

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

layout: docs

page_title: Manual license utilization reporting

description: >-

Manual license utilization reporting allows you to export, review, and send license utilization data to HashiCorp through the CLI or HCP Web Portal.

Manual license utilization reporting

@include 'alerts/enterprise-only.mdx'

Manual license utilization reporting allows you to export, review, and send license utilization data to HashiCorp via the CLI or HCP Web Portal. Use these reports to understand how much more you can deploy under your current contract, protect against overutilization, and budget for predicted consumption. Manual reporting shares the minimum data required to validate license utilization as defined in our contracts. The reports consist of mostly computed metrics and will never contain Personal Identifiable Information (PII) or other sensitive information.

Manual license utilization shares the same data as automated license utilization but is more time consuming. Unless you are running in an air-gapped environment or have another reason to report data manually, we strongly recommend using automated reporting instead. If you have disabled automated license reporting, you can re-enable it by reversing the opt-out process described in the [documentation](/vault/docs/enterprise/license/utilization-reporting#opt-out).

If you are considering manual reporting because you're worried about your data, we strongly recommend that you review the [example payloads](#data-file-content), which are the same for automated and manual reporting. If you have further concerns with any of the automatically-reported data please bring them to your account manager before opting out of automated reporting in favor of manual reporting.

How to manually send data reports

Generate a data bundle

Data bundles include collections of JSON snapshots that contain license utilization information.

1. Login into your [cluster node](/vault/tutorials/cloud/vault-access-cluster).

1. Run

this CLI command to generate a data bundle:

```
```shell-session
$ vault operator utilization
```
```

By default, the bundle will include all historical snapshots.

You can provide context about the conditions under which the report was generated and submitted by providing a comment. This optional comment will not be included in the license utilization bundle, but will be included in the Vault server logs.

****Example:****

```
``shell-session
$ vault operator utilization -message=Change Control 654987 \
  -output=/utilization/reports/latest.json
``
```

This command will export all the persisted snapshots into a bundle. The message Change Control 654987 will not be included in the bundle but will be included in Vault server logs. The `-output`` flags specifies the output location of the JSON bundle.

****Available command flags:****

- `-message` `(string:)`` - Provide context about the conditions under which the report was generated and submitted. This message is not included in the license utilization bundle but will be included in the vault server logs. (optional)

- `-today-only` `(bool: false)`` - To include only today's snapshot, no historical snapshots. If no snapshots were persisted in the last 24 hrs, it takes a snapshot and exports it to a bundle. (optional)

- `-output` `(string:)`` - Specifies the output path for the bundle. Defaults to a time-based generated file name. (optional)

Send the data bundle to HashiCorp

1. Go to <https://portal.cloud.hashicorp.com/license-utilization/reports/create>
1. Click on ****Choose files****, or drop your file(s) into the container.

a. If the upload succeeded, the HCP user interface will change the file status to ****Uploaded**** in green.

b. If the upload failed, the file status will say ****Failed**** in red, and will include error information.

If the upload fails make sure you haven't modified the file signature. If the error persists, please contact your account representative.

Enable manual reporting

Upgrade to a release that supports manual license utilization reporting. These

releases include:

- Vault Enterprise 1.16.0 and later
- Vault Enterprise 1.15.6 and later
- Vault Enterprise 1.14.10 and later

Configuration

Administrators can manage disk space for storing snapshots by defining the number of days snapshots can be retained.

```
```hcl
reporting {
 snapshot_retention_time = "2400h"
}
```
```

The default retention period is 400 days.

Data file content

<CodeBlockConfig hideClipboard>

```
```json
{
 "snapshot_version": 2,
 "id": "0001JWAY00BRF8TEXC9CVRHBAC",
 "timestamp": "2024-02-08T16:55:28.085215-08:00",
 "schema_version": "2.0.0",
 "product": "vault",
 "process_id": "01HP5NJS21HN50FY0CBS0SYGCH",
 "metrics": {
 "clientcount.current_month_estimate.type.entity": {
 "key": "clientcount.current_month_estimate.type.entity",
 "value": 20,
 "mode": "write"
 },
 "clientcount.current_month_estimate.type.nonentity": {
 "key": "clientcount.current_month_estimate.type.nonentity",
 "value": 11,
 "mode": "write"
 },
 "clientcount.current_month_estimate.type.secret_sync": {
 "key": "clientcount.current_month_estimate.type.secret_sync",
 "value": 0,
 "mode": "write"
 }
 }
}
```



```

"clientcount.previous_month_complete.type.entity": {
 "key": "clientcount.previous_month_complete.type.entity",
 "value": 0,
 "mode": "write"
},
"clientcount.previous_month_complete.type.nonentity": {
 "key": "clientcount.previous_month_complete.type.nonentity",
 "value": 0,
 "mode": "write"
},
"clientcount.previous_month_complete.type.secret_sync": {
 "key": "clientcount.previous_month_complete.type.secret_sync",
 "value": 0,
 "mode": "write"
}
},
"product_version": "1.16.0+ent",
"license_id": "7d68b16a-74fe-3b9f-a1a7-08cf461fff1c",
"checksum": 6861637915450723051,
"metadata":
{
 "billing_start": "2023-05-04T00:00:00Z",
 "cluster_id": "16d0ff5b-9d40-d7a7-384c-c9b95320c60e"
}
}

```

</CodeBlockConfig>

---

layout: docs

page\_title: Vault Enterprise License

description: An overview of license.

---

# Vault license

@include 'alerts/enterprise-only.mdx'

Licenses and EULA enhancements have been introduced in Vault 1.8 release. Please refer to the [FAQ](/vault/docs/enterprise/license/faq) for common questions concerning these changes.

The [Install a HashiCorp Enterprise License](/vault/tutorials/enterprise/hashicorp-enterprise-license) tutorial provides the instruction to load your Vault license.

License text copyright (c) 2020 MariaDB Corporation Ab, All Rights Reserved.

"Business Source License" is a trademark of MariaDB Corporation Ab.

Parameters

Licensors: HashiCorp, Inc.

Licensed Work: Vault Version 1.15.0 or later. The Licensed Work is (c) 2024 HashiCorp, Inc.

Additional Use Grant: You may make production use of the Licensed Work, provided Your use does not include offering the Licensed Work to third parties on a hosted or embedded basis in order to compete with HashiCorp's paid version(s) of the Licensed Work. For purposes of this license:

A "competitive offering" is a Product that is offered to third parties on a paid basis, including through paid support arrangements, that significantly overlaps with the capabilities of HashiCorp's paid version(s) of the Licensed Work. If Your

Product is not a competitive offering when You first make it generally available, it will not become a competitive offering later due to HashiCorp releasing a new version of the Licensed Work with additional capabilities. In addition, Products that are not provided on a paid basis are not competitive.

"Product" means software that is offered to end users to manage in their own environments or offered as a service on a hosted basis.

"Embedded" means including the source code or executable code from the Licensed Work in a competitive offering. "Embedded" also means packaging the competitive offering in such a way that the Licensed Work must be accessed or downloaded for the

competitive offering to operate.

Hosting or using the Licensed Work(s) for internal purposes within an organization is not considered a competitive offering. HashiCorp considers your organization to include all of your affiliates under common control.

For binding interpretive guidance on using HashiCorp products under the Business Source License, please visit our FAQ. (<https://www.hashicorp.com/license-faq>)

Change Date: Four years from the date the Licensed Work is published.

Change License: MPL 2.0

For information about alternative licensing arrangements for the Licensed Work, please contact [licensing@hashicorp.com](mailto:licensing@hashicorp.com).

Notice

Business Source License 1.1

## Terms

The Licensor hereby grants you the right to copy, modify, create derivative works, redistribute, and make non-production use of the Licensed Work.

The

Licensor may make an Additional Use Grant, above, permitting limited production use.

Effective on the Change Date, or the fourth anniversary of the first publicly available distribution of a specific version of the Licensed Work under this License, whichever comes first, the Licensor hereby grants you rights under the terms of the Change License, and the rights granted in the paragraph above terminate.

If your use of the Licensed Work does not comply with the requirements currently in effect as described in this License, you must purchase a commercial license from the Licensor, its affiliated entities, or authorized resellers, or you must refrain from using the Licensed Work.

All copies of the original and modified Licensed Work, and derivative works of the Licensed Work, are subject to this License. This License applies separately for each version of the Licensed Work and the Change Date may vary for each version of the Licensed Work released by Licensor.

You must conspicuously display this License on each original or modified copy of the Licensed Work. If you receive the Licensed Work in original or modified form from a third party, the terms and conditions set forth in this License apply to your use of that work.

Any use of the Licensed Work in violation of this License will automatically terminate your rights under this License for the current and all other versions of the Licensed Work.

This License does not grant you any right in any trademark or logo of Licensor or its affiliates (provided that you may use a trademark or logo of Licensor as expressly required by this License).

TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE LICENSED WORK IS PROVIDED ON AN "AS IS" BASIS. LICENSOR HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, AND TITLE.

MIT License

Copyright (c) 2021 Taishi Kasuga

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

layout: docs

page\_title: Frequently Asked Questions (FAQ)

description: An overview of license.

---

## # License FAQ

This FAQ section is for license changes and updates introduced for Vault Enterprise.

- [Q: How do the license termination changes affect upgrades?](#q-how-do-the-license-termination-changes-affect-upgrades)
- [Q: What impact on upgrades do the license termination behavior changes pose?](#q-what-impact-on-upgrades-do-the-license-termination-behavior-changes-pose)
- [Q: Will these license changes impact HCP Vault?](#q-will-these-license-changes-impact-hcp-vault)
- [Q: Do these license changes impact all Vault customers/licenses?](#q-do-these-license-changes-impact-all-vault-customers-licenses)
- [Q: What is the product behavior change introduced by the licensing changes?](#q-what-is-the-product-behavior-change-introduced-by-the-licensing-changes)
- [Q: How will Vault behave at startup when a license expires or terminates?](#q-how-will-vault-behave-at-startup-when-a-license-expires-or-terminates)
- [Q: What is the impact on evaluation licenses due to this change?](#q-what-is-the-impact-on-evaluation-licenses-due-to-this-change)
- [Q: Are there any changes to existing methods for manual license loading (API or CLI)?](#q-are-there-any-changes-to-existing-methods-for-manual-license-loading-api-or-cli)
- [Q: Is there a grace period when evaluation licenses expire?](#q-is-there-a-grace-period-when-evaluation-licenses-expire)
- [Q: Are the license files locked to a specific cluster?](#q-are-the-license-files-locked-to-a-specific-cluster)
- [Q: Will a EULA check happen every time a Vault restarts?](#q-will-a-eula-check-happen-every-time-a-vault-restarts)
- [Q: What scenarios should a customer plan for due to these license changes?](#q-what-scenarios-should-a-customer-plan-for-due-to-these-license-changes)

- [Q: What is the migration path for customers who want to migrate from their existing license-as-applied-via-the-CLI flow to the license on disk flow?](#q-what-is-the-migration-path-for-customers-who-want-to-migrate-from-their-existing-license-as-applied-via-the-cli-flow-to-the-license-on-disk-flow)

-

[Q: What is the path for customers who want to downgrade/rollback from Vault 1.11 or later (auto-loaded license mandatory) to a pre-Vault 1.11 (auto-loading not mandatory, stored license supported)?](#q-what-is-the-path-for-customers-who-want-to-downgrade-rollback-from-vault-1-11-or-later-auto-loaded-license-mandatory-to-a-pre-vault-1-11-auto-loading-not-mandatory-stored-license-supported)

- [Q: Is there a limited time for support of licenses that are in storage?](#q-is-there-a-limited-time-for-support-of-licenses-that-are-in-storage)

- [Q: What are the steps to upgrade from one autoloading license to another autoloading license?](#q-what-are-the-steps-to-upgrade-from-one-autoloading-license-to-another-autoloading-license)

- [Q: What are the Vault ADP module licensing changes introduced in 1.8?](#q-what-are-the-vault-adp-module-licensing-changes-introduced-in-1-8)

-

[Q: How can the new ADP modules be purchased and what features are customer entitled to as part of that purchase?](#q-how-can-the-new-adp-modules-be-purchased-and-what-features-are-customer-entitled-to-as-part-of-that-purchase)

- [Q: What is the impact to customers based on these ADP module licensing changes?](#q-what-is-the-impact-to-customers-based-on-these-adp-module-licensing-changes)

### Q: what is the product behavior change introduced by the licensing changes?

Per the [feature deprecation plans](/vault/docs/deprecation), Vault will no longer support licenses in storage. An [auto-loaded license](/vault/docs/enterprise/license/autoloading) must be used instead. If you are using stored licenses, you must migrate to auto-loaded licenses prior to upgrading to Vault 1.11

Vault 1.12 will also introduce different termination behavior for evaluation licenses versus non-evaluation licenses. An evaluation license will include a 30-day trial period after which a running Vault server will terminate. Vault servers using a non-evaluation license will not terminate.

### Q: how do the license termination changes affect upgrades?

Vault 1.12 will introduce changes to the license termination behavior. Upgrades when using expired licenses will now be limited.

Vault will not startup if the build date of the binary is after the expiration date of a license. License expiration date and binary build date compatibility can be verified using the [Check for Autoloading License](/vault/docs/commands/operator/diagnose#check-for-autoloading-license) check performed by the ``vault operator diagnose`` command.

The build date of a binary can also be found using the [vault version](/vault/docs/commands/version#version) command.

A user can expect the following to occur based on the following scenarios:

**\*\*Evaluation or non-evaluation license is valid:\*\***

Vault will start normally

**\*\*Evaluation or non-evaluation license is expired, binary build date before license expiry**

date:\*\*

Vault will start normally

**\*\*Evaluation or non-evaluation license is expired, binary build date \_after\_ license expiry date:\*\***

Vault will not start

**\*\*Evaluation license is terminated:\*\***

Vault will not start independent of the binary's build date

**\*\*Non-evaluation license is terminated, binary build date \_before\_ license expiry date:\*\***

Vault will start normally

**\*\*Non-evaluation license is terminated, binary build date \_after\_ license expiry date:\*\***

Vault will not start

The Vault support team can issue you a temporary evaluation license to allow for security upgrades if your license has expired.

### Q: will these license changes impact HCP Vault?

No, these changes will not impact HCP Vault.

### Q: do these license changes impact all Vault customers/licenses?

| Customer/licenses                                                                                                              | Impacted? |
|--------------------------------------------------------------------------------------------------------------------------------|-----------|
| -----                                                                                                                          | -----     |
| ENT binaries (evaluation or non-evaluation downloaded from<br>[releases.hashicorp.com](https://releases.hashicorp.com/vault/)) | Yes       |
| Business-Source (BSL)                                                                                                          | No        |

### Q: what is the product behavior change introduced by the licensing changes?

With Vault 1.11, the use of an [auto-loaded license](/vault/docs/enterprise/license/autoloading) is required for Vault to start successfully.

### Q: how will Vault behave at startup when a license expires or terminates?

When a license expires, Vault continues to function until the license terminates. This behavior exists today and remains unchanged in Vault 1.11. The grace period, defined as the time between license expiration and license termination, is one day for evaluation licenses (as of 1.8), and ten years for non-evaluation licenses.

Customers must provide a

valid license before the grace period expires. This license is required to be [auto-loaded](/vault/docs/enterprise/license/autoloading). When license terminates (upon grace period expiry), Vault will seal itself and customers will need a valid license in order to successfully bring-up Vault. If a valid license was not installed after license expiry, customers will need to provide one, and this license will need to be auto-loaded.

Vault 1.12 changes the license expiration and termination behavior. Evaluation licenses include a 30-day trial period

after which a running Vault server will terminate. Non-evaluation licenses, however, will no longer terminate. When a non-evaluation license expires, Vault will continue to function but upgrades will be limited. The build date of the upgrade binary must be before the expiration date of the license.

Vault will not start when attempting to use an expired license and binary with a build date *\_after\_* the license expiration date. Attempting to [reload](/vault/api-docs/system/config-reload#reload-license-file)

an expired license will result in an error if the build date of the running Vault server is *\_after\_* the license expiration date.

License expiration date and binary build date compatibility can be verified using the [Check for Autoloaded License](/vault/docs/commands/operator/diagnose#check-for-autoloaded-license) check performed by the `vault operator diagnose` command. The build date of a binary can also be found using the [vault version](/vault/docs/commands/version#version) command.

### Q: what is the impact on evaluation licenses due to this change?

As of Vault 1.8, any Vault cluster deployed must have a valid [auto-loaded](/vault/docs/enterprise/license/autoloading) license.

Vault 1.12 introduces [expiration and termination behavior changes](#q-how-will-vault-behave-at-startup-when-a-license-expires-or-terminates) for non-evaluation licenses. Evaluation licenses will continue to have a 1-day grace period upon license

expiry after which they will terminate. Vault will seal itself and shutdown once an evaluation license terminates.

### Q: are there any changes to existing methods for manual license loading (API or CLI)?

The ["/sys/license"](/vault/api-docs/v1.10.x/system/license#install-license) and ["/sys/license/signed"](/vault/api-docs/v1.10.x/system/license#read-signed-license) endpoints have been removed as of Vault 1.11. With that said, it is no longer possible to provide a license via the `"/sys/license` endpoint. License [auto-loading](/vault/docs/enterprise/license/autoloading) must be used instead.

The ["/sys/config/reload/license"](/vault/api-docs/system/config-reload#reload-license-file) endpoint can be used to reload an auto-loaded license provided as a path via an environment variable or configuration.

### Q: is there a grace period when evaluation licenses expire?

Evaluation licenses have a 1-day grace period. The grace period is the time until the license expires. Upon expiration,

Vault will seal and will require a valid license to unseal and function properly.

### Q: are the license files locked to a specific cluster?

The changes to licenses apply to all nodes in a cluster. The license files are not locked to a cluster, but are independently applied to the appropriate clusters.

### Q: will a EULA check happen every time a Vault restarts?

Yes, starting with Vault 1.8, ENT binaries will be subjected to a EULA check. The release of Vault 1.8 introduces the EULA check for evaluation licenses (non-evaluation licenses are evaluated with a EULA check during contractual agreement) .

Although the agreement to a EULA occurs only once (when the user receives their license), Vault will check for the presence of a valid license every time a node is restarted.

Starting Vault 1.11, when customers deploy new Vault clusters, or upgrade existing Vault clusters, a valid [auto-loaded](/vault/docs/enterprise/license/autoloading) license must exist for the upgrade to be successful.

###

Q: what scenarios should a customer plan for due to these license changes?

- **New Cluster Deployment**: When a customer deploys new clusters to Vault 1.11 or later, a valid license must exist to successfully deploy. This valid license must be on-disk ([auto-loaded](/vault/docs/enterprise/license/autoloading)).

- **Eventual Migration**: Vault 1.11 removes support for in-storage licenses. Migrating to an auto-loaded license is required for Vault to start successfully using version 1.11 or greater. Pre-existing license storage entries will be automatically removed from storage upon startup.

### Q: what is the migration path for customers who want to migrate from their existing license-as-applied-via-the-CLI flow to the license on disk flow?

If a Vault cluster using a stored license is planned to be upgraded to Vault 1.11 or greater, the operator must migrate to using an auto-loaded license. The [vault license get -signed](/vault/docs/v1.10.x/commands/license/get) command

(or underlying [sys/license/signed](/vault/api-docs/v1.10.x/system/license#read-signed-license) endpoint) can be used to retrieve the license from storage in a running cluster.

It is not necessary to remove the stored license entry. That will occur automatically upon startup in Vault 1.11 or greater. Prior to completing the [recommended upgrade steps](/vault/docs/upgrading), perform the following to ensure your license is properly configured:

1. Use the command `vault license get -signed` to retrieve the license from storage of your running cluster.
2. Put the license on disk
3. Configure license auto-loading by specifying the [license\_path](/vault/docs/configuration#license\_path) config option or setting the [VAULT\_LICENSE](/vault/docs/commands#vault\_license) or [VAULT\_LICENSE\_PATH](/vault/docs/commands#vault\_license\_path) environment variable.

### Q: what is the path for customers who want to downgrade/rollback from Vault 1.11 or later (auto-loaded license mandatory) to a pre-Vault 1.11 (auto-loading not mandatory, stored license supported)?

The downgrade procedure remains the same for Vault customers who are currently on Vault 1.11 or later, have a license installed via auto-loading, and would like to downgrade their cluster to a pre-1.11 version. Please refer to the [upgrade procedures](/vault/tutorials/standard-procedures/sop-upgrade) that remind the customers that they must take a snapshot before the upgrade. Customers will need to restore their version from the snapshot, apply the pre-1.11 enterprise binary version they wish to roll back, and bring up the clusters.

### Q: is there a limited time for support of licenses that are in storage?

The support of licenses installed by alternative means often leads to difficulties providing the appropriate support.



To provide the support expected by our customers, as we have announced in [Vault feature deprecations and plans](/vault/docs/deprecation) we are removing support for licenses in storage with

Vault 1.11. This implies licensing endpoints that deal with licenses in storage will be removed, and Vault will no longer check for valid licenses in storage. This change requires that all customers have [auto-loaded](/vault/docs/enterprise/license/autoloading) licenses to upgrade to 1.11(+) successfully.

### Q: what are the steps to upgrade from one autoloaded license to another autoloaded license?

Follow these steps to migrate from one autoloaded license to another autoloaded license.

1. Use the [`vault license inspect`](/vault/docs/commands/license/inspect) command to compare the new license against the output of the [`vault license get`](/vault/docs/commands/license/get) command. This is to ensure that you have the correct license.

1. Backup the old license file in a safe location.

1. Replace the old license file on each Vault server with the new one.

1. Invoke the [reload command](/vault/api-docs/system/config-reload#reload-license-file) on each individual Vault server, starting

with the standbys and doing the leader last. Invoking in this manner reduces possible disruptions if something was performed incorrectly with the above steps. You can either use the [reload command](/vault/api-docs/system/config-reload#reload-license-file) or send a SIGHUP.

1. On each node, ensure that the new license is in use by using the [`vault license get`](/vault/docs/commands/license/get) command and/or checking the logs.

# ADP licensing

This FAQ section is for the Advanced Data Protection (ADP) license changes introduced in Vault Enterprise 1.8.

### Q: what are the Vault ADP module licensing changes introduced in 1.8?

As of Vault Enterprise 1.8, the functionality formerly sold as the Vault ADP module is now separated between two new modules:

**\*\*ADP-KM\*\*** includes:

- [Key Management Secrets Engine (KMSE)](/vault/docs/secrets/key-management)
- [Key Management Interoperability (KMIP)](/vault/docs/secrets/kmip)
- [MSSQL Transparent Data Encryption (TDE)](https://www.hashicorp.com/blog/enabling-transparent-data-encryption-for-microsoft-sql-with-vault)

**\*\*ADP-Transform\*\***

includes:

- [Transform Secrets Engine (TSE)](/vault/docs/secrets/transform)

### Q: how can the new ADP modules be purchased and what features are customer entitled to as part of that purchase?

**\*\*ADP-KM includes\*\***:

- This is the first Vault Enterprise module that can be purchased standalone. This means it can be purchased without the purchase of a Vault Enterprise Standard license.
- ADP-KM still requires a Vault Enterprise binary. The Vault Enterprise Standard license is automatically included with the ADP-KM module, but customers are contractually prohibited from using any features besides those in Vault Community Edition and ADP-KM (KMSE and KMIP).

**\*\*ADP-Transform includes\*\*:**

- This module cannot be purchased as a standalone. It requires a Vault Enterprise binary, and customers must purchase the base Vault Enterprise Standard license (at least) to use the corresponding Enterprise features.

-

The ADP-Transform SKU can be applied as an add-on. This workflow is similar to the consolidated ADP SKU.

### Q: what is the impact to customers based on these ADP module licensing changes?

Customers need to be aware of the following as a result of these changes:

- **\*\*New customers\*\*** may choose to purchase either or both of these modules. The old (consolidated) module is not available to them as an option.
- **\*\*Existing customers\*\*** may continue with the consolidated Vault ADP module uninterrupted. They will only be converted to one or both new ADP modules the next time they make a change to their licensing details (i.e. contract change).

# Copyright (c) HashiCorp, Inc.

# SPDX-License-Identifier: BUSL-1.1

```
variable "file_name" {}
```

```
output "license" {
 value = file(var.file_name)
}
```

---

layout: docs

page\_title: License Autoloading

description: An overview of license autoloading.

---

# License autoloading

@include 'alerts/enterprise-only.mdx'

Prior to Vault 1.8, Vault Enterprise would be licensed using special binaries that contained embedded licenses, or via a license written into Vault storage using the [POST sys/license API](/vault/api-docs/system/license#install-license).

As of Vault 1.8 those options still exist but are deprecated, and the recommended mechanism for managing licenses is called License Autoloading. New clusters are required to use the autoloading method.

License Autoloading can be done using one of these mechanisms, in decreasing order of priority (i.e. `VAULT\_LICENSE` takes precedence over `VAULT\_LICENSE\_PATH`, which takes precedence over `license\_path` in config.)

- [VAULT\_LICENSE environment variable](/vault/docs/commands#vault\_license)
- [VAULT\_LICENSE\_PATH environment variable](/vault/docs/commands#vault\_license\_path)
- [license\_path in config](/vault/docs/configuration#license\_path)

Nodes within a cluster should be consistently licensed. All nodes in a cluster should use autoloading, or none should. If autoloading is used, all nodes should use the same license string. Warnings will be written to the log when there are discrepancies.

If autoloading is used, any existing stored license will be ignored. The presence of a stored license in conjunction with an autoloaded license will also result in logged warnings, which may require a service restart to clear.

---

layout: docs

page\_title: license inspect - Command

description: |-

The "license inspect" command describes the contents of a signed license string.

---

# license inspect

The `license inspect` command describes the contents of a signed license string.

## Examples

Describe license in the environment (either `VAULT\_LICENSE` for a raw value or `VAULT\_LICENSE\_PATH` for the path to a file containing the value):

```
```shell-session
```

```
$ vault license inspect
```

```
Source: /Users/ncc/vaults/license
```

```
Product: vault
```

```
License ID: 060d7820-fa59-f95c-832b-395db0aeb9ba
```

```
Customer ID: 5401ae3b-136b-3e47-3b43-432927fa03e7
```

```
Installation ID: *
```

```
Issue Time: 2021-05-17 19:33:47.93981585 +0000 UTC
```

```
Start Time: 2021-05-17 00:00:00 +0000 UTC
```

```
Expiration Time: 2022-05-17 23:59:59.999 +0000 UTC
```

```
Termination Time: 2031-05-17 23:59:59.999 +0000 UTC
```

```
{ "license_id": "060d7820-fa59-f95c-832b-395db0aeb9ba", "customer_id": "5401ae3b-136b-3e47-3b43-
```

```
432927fa03e7", "installation_id": "*", "issue_time": "2021-05-17T19:33:47.93981585Z", "start_time": "2021-05-
```

```
17T00:00:00Z", "expiration_time": "2022-05-17T23:59:59.999Z", "flags": { "modules": [ "multi-dc-scale", "governance-
```

```
policy","advanced-data-protection"]},"features":["HSM","Performance
Replication","DR Replication","MFA","Sentinel","Seal Wrapping","Control Groups","Performance
Standby","Namespaces","KMIP","Entropy Augmentation","Transform Secrets Engine","Lease Count Quotas","Key
Management Secrets Engine","Automated Snapshots"],"performance_standby_count":9999}
```

License is valid

```

Describe license at given path:

```
```shell-session
```

```
$ vault license inspect ~/vaults/license
```

```
Source: /Users/ncc/vaults/license
```

```
Product: vault
```

```
License ID: 060d7820-fa59-f95c-832b-395db0aeb9ba
```

```
Customer ID: 5401ae3b-136b-3e47-3b43-432927fa03e7
```

```
Installation ID: *
```

```
Issue Time: 2021-05-17 19:33:47.93981585 +0000 UTC
```

```
Start Time: 2021-05-17 00:00:00 +0000 UTC
```

```
Expiration Time: 2022-05-17 23:59:59.999 +0000 UTC
```

```
Termination Time: 2031-05-17 23:59:59.999 +0000 UTC
```

```
{"license_id":"060d7820-fa59-f95c-832b-395db0aeb9ba","customer_id":"5401ae3b-136b-3e47-3b43-
432927fa03e7","installation_id":"*","issue_time":"2021-05-17T19:33:47.93981585Z","start_time":"2021-05-
17T00:00:00Z","expiration_time":"2022-05-17T23:59:59.999Z","flags":{"modules":["multi-dc-scale","governance-
policy","advanced-data-protection"]},"features":["HSM","Performance
Replication","DR Replication","MFA","Sentinel","Seal Wrapping","Control Groups","Performance
Standby","Namespaces","KMIP","Entropy Augmentation","Transform Secrets Engine","Lease Count Quotas","Key
Management Secrets Engine","Automated Snapshots"],"performance_standby_count":9999}
```

License is valid

```

## ## Usage

The following flags are available in addition to the [standard set of flags](/vault/docs/commands) included on all commands.

### ### Command options

- `-installation-id` (string: "")`` - Installation ID to validate the license against

### ### Enterprise licenses

In versions 1.2.6, 1.3.9, 1.4.5, and 1.5.2, the enterprise licenses were not incorporated correctly into the build and we have issued patch releases (x.y.z.1) for enterprise customers containing the proper license. As the previous builds were both not working and causing confusion, we have removed the binaries.

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

package license

// Features is a bitmask of feature flags

type Features uint

const FeatureNone Features = 0

func (f Features) HasFeature(flag Features) bool {

return false

}

### vault.core.license.expiration\_time\_epoch ((#vault-core-license-expiration\_time\_epoch))

Metric type | Value | Description

----- | ----- | -----

gauge | timestamp | Epoch time (seconds since 1970-01-01) at which the license will expire

---

layout: docs

page\_title: license - Command

description: |-

The "license" command groups subcommands for interacting with Vault licenses.

---

# license

The `license` command groups subcommands for interacting with Vault licenses

For more information, please see the [license documentation](/vault/docs/enterprise/license)

## Examples

Query current license in use by a node:

```shell-session

\$ vault license get

Key Value

expiration_time 2022-05-17T23:59:59.999Z

features [HSM Performance Replication DR Replication MFA Sentinel Seal Wrapping Control Groups
Performance Standby Namespaces KMIP Entropy Augmentation Transform Secrets Engine Lease Count Quotas
Key Management Secrets Engine Automated Snapshots]

license_id 060d7820-fa59-f95c-832b-395db0aeb9ba

performance_standby_count 9999

start_time 2021-05-17T00:00:00Z

```

Inspect a license string:

```

```shell-session
$
  vault license inspect ~/vaults/license
Source: /Users/ncc/vaults/license
Product: vault
License ID: 060d7820-fa59-f95c-832b-395db0aeb9ba
Customer ID: 5401ae3b-136b-3e47-3b43-432927fa03e7
Installation ID: *
Issue Time: 2021-05-17 19:33:47.93981585 +0000 UTC
Start Time: 2021-05-17 00:00:00 +0000 UTC
Expiration Time: 2022-05-17 23:59:59.999 +0000 UTC
{"license_id":"060d7820-fa59-f95c-832b-395db0aeb9ba","customer_id":"5401ae3b-136b-3e47-3b43-432927fa03e7","installation_id":"*","issue_time":"2021-05-17T19:33:47.93981585Z","start_time":"2021-05-17T00:00:00Z","expiration_time":"2022-05-17T23:59:59.999Z","flags":{"modules":["multi-dc-scale","governance-policy","advanced-data-protection"]},"features":["HSM","Performance Replication","DR Replication","MFA","Sentinel","Seal Wrapping","Control Groups","Performance Standby","Namespaces","KMIP","Entropy Augmentation","Transform Secrets Engine","Lease Count Quotas","Key Management Secrets Engine","Automated Snapshots"],"performance_standby_count":9999}

License
is valid

```

```

## Usage

```

```text
Usage: vault license <subcommand> [options] [args]

```

...

Subcommands:

```

  get    Query Vault's license
  inspect View the contents of a license string
```

```

For more information, examples, and usage about a subcommand, click on the name of the subcommand in the sidebar.

## 1.72 x-time 0.11.0

### 1.72.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.73 x-time-rate 0.11.0

### 1.73.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2015 The Go Authors. All rights reserved.

Found in path(s):

\* /time-refs-tags-v0-11-0-rate-tar-gz/rate\_test.go  
\* /time-refs-tags-v0-11-0-rate-tar-gz/rate.go

No license file was found, but licenses were detected in source scan.

// Copyright 2022 The Go Authors. All rights reserved.

Found in path(s):

\* /time-refs-tags-v0-11-0-rate-tar-gz/sometimes\_test.go  
\* /time-refs-tags-v0-11-0-rate-tar-gz/sometimes.go

## 1.74 go-secure-stdlib-parseutil 0.2.0

## 1.74.1 Available under license :

Copyright (c) 2020 HashiCorp, Inc.

Mozilla Public License, version 2.0

### 1. Definitions

#### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

#### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

#### 1.3. "Contribution"

means Covered Software of a particular Contributor.

#### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

#### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

#### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

#### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. "License"



means this document.

#### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)  
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

### 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

b. You may distribute such Executable Form under the terms of this

License, or sublicense it under different terms,  
provided that the  
license for the Executable Form does not attempt to limit or alter the  
recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4.

### Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a

text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward.  
Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.  
Copyright (c) 2024 HashiCorp, Inc.

Mozilla Public License, version 2.0

### 1. Definitions

#### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

#### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

#### 1.3. "Contribution"

means Covered Software of a particular Contributor.

#### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

#### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

#### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

#### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. "License"

means this document.

#### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the



rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)  
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4.

### Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any

character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward.  
Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

#### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

##### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

##### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

## 1.75 [github.com/googleapis/enterprise-certificate-proxy](https://github.com/googleapis/enterprise-certificate-proxy) 0.3.6

### 1.75.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner



or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.76 enterprise-certificate-proxy 0.3.6

### 1.76.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership  
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.77 [github.com/hashicorp/go-secure-stdlib/parseutil](https://github.com/hashicorp/go-secure-stdlib/parseutil) 0.2.0

## 1.77.1 Available under license :

Copyright (c) 2020 HashiCorp, Inc.

Mozilla Public License, version 2.0

### 1. Definitions



### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

### 1.3. "Contribution"

means Covered Software of a particular Contributor.

### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

### 1.8. "License"

means this document.

### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the

rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)  
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4.

### Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any

character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward.  
Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

#### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.  
Copyright (c) 2024 HashiCorp, Inc.

Mozilla Public License, version 2.0

### 1. Definitions

#### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

#### 1.2. "Contributor Version"



means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

### 1.3. "Contribution"

means Covered Software of a particular Contributor.

### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

### 1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

### 1.8. "License"

means this document.

### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to,

deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

### 2. License Grants and Conditions

#### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for

sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of

the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4.

### Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by

some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section

2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of

incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward.  
Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

#### 10.4. Distributing Source Code Form that is Incompatible With Secondary

Licenses If You choose to distribute Source

Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at

<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

## 1.78 protobuf-go 1.36.6

### 1.78.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.



THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.79 protobuf 1.36.6

### 1.79.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.80 logrus 1.9.3

## 1.80.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.81 perks 1.0.1

## 1.81.1 Available under license :

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.82 prometheus-client 1.22.0

### 1.82.1 Available under license :

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Prometheus instrumentation library for Go applications

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>

<https://github.com/beorn7/perks>

Copyright 2013-2015 Blake Mizerany, Björn Rabenstein

See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format

<http://github.com/golang/protobuf/>

Copyright 2010 The Go Authors

See source code for license details.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.83 clientmodel 0.6.2

## 1.83.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.



"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# 1.84 prometheus-client-model 0.6.2

## 1.84.1 Available under license :

Data model artifacts for Prometheus.  
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership  
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but  
not limited to compiled object code, generated documentation,  
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf



of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.85 [github.com/go-jose/go-jose/v4](https://github.com/go-jose/go-jose/v4) 4.1.0

## 1.85.1 Available under license :

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.86 cloud.google.com/go/auth/oauth2adapt 0.2.8

## 1.86.1 Available under license :

```
// Copyright 2024 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
```

```

//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{

```

```

// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#ListLicensesRequest.
}
for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {

```



```

// TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := procurement.NewLicenseManagementRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseManagementClient_Assign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
 }
 resp, err := c.Assign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

//
TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
 }
 it := c.EnumerateLicensedUsers(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 //
 TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
 }
}

func ExampleLicenseManagementClient_GetLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.

```

```

// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRe
 quest.
}
resp, err := c.GetLicensePool(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_Unassign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UnassignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest.
 }
 resp,
 err := c.Unassign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_UpdateLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UpdateLicensePoolRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
 }
 resp, err := c.UpdateLicensePool(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 //
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }

```

```

}
resp, err := c.GetOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

// TODO: Use client.
_ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.

```

```

// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseRequest.
}
resp,
err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
 }
 resp, err := c.GetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

defer c.Close()

req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#InsertLicenseRequest.
}
op, err := c.Insert(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template
 // only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#ListLicensesRequest.
 }
 it := c.List(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,

```



```

// you can do so by casting the `Response` as below.
// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*compute.LicensesListResponse)
}
}

func
ExampleLicensesClient_SetIamPolicy() {
ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
// TODO: Handle error.
}
defer c.Close()

req := &compute.SetIamPolicyLicenseRequest{
// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute.LicensesRESTClient#SetIamPolicyLicenseRequest.
}
resp, err := c.SetIamPolicy(ctx, req)
if err != nil {
// TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
ctx := context.Background()
// This snippet has been automatically
// generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
// TODO: Handle error.
}
defer c.Close()

req := &compute.TestIamPermissionsLicenseRequest{

```

```
// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
}
resp, err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

package compute

import (

"bytes"

"context"

"fmt"

"log/slog"

"math"

"net/http"

"net/url"

"time"

computepb "cloud.google.com/go/compute/apiv1/computepb"

gax "github.com/googleapis/gax-go/v2"

"google.golang.org/api/iterator"

"google.golang.org/api/option"

"google.golang.org/api/option/internaloption"

httptransport "google.golang.org/api/transport/http"

"google.golang.org/grpc"

"google.golang.org/protobuf/encoding/protojson"

"google.golang.org/protobuf/proto"

)

var

```
newLicensesClientHook clientHook
```

```
// LicensesCallOptions contains the retry settings for each method of LicensesClient.
```

```
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
}
```

```
func defaultLicensesRESTCallOptions() *LicensesCallOptions {
 return &LicensesCallOptions{
 Delete: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 GetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}
```



```

gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
},
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
}),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

//
internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse,
 error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.

```

```

CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection
// to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty
// if no such policy or resource exists. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

```

```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *compute.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that
belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a
list of publicly-available licenses,
use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution
This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *compute.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *compute.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*compute.Policy, error) {
 return c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners
who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*compute.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*compute.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

```

```

// Points back to the CallOptions field of the containing LicensesClient
CallOptions **LicensesCallOptions

logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx
context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
if err != nil {
return nil, err
}

callOpts := defaultLicensesRESTCallOptions()
c := &licensesRESTClient{
endpoint: endpoint,
httpClient: httpClient,
CallOptions: &callOpts,
logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
option.WithHTTPClient(httpClient),
option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {
return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
internaloption.WithDefaultUniverseDomain("googleapis.com"),
internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
internaloption.WithDefaultScopes(DefaultAuthScopes()...),
}
}

```

```

 internaloption.EnableNewAuthLibrary(),
}
}

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN")
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient
 with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {
 return err
 }
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
}

```

```

}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client
and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf,
resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}

```

```

return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders,
hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {
 return nil, e
 }
}

```

```

}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource
// exists. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace
// images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy", req.GetProject(),
req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds,
"Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")

```



```

 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended
// for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx,
 hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)

```

```

unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// List retrieves the
// list of licenses available in the specified project. This method does not get any licenses that belong to other projects,
// including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available
// licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
...gax.CallOption) *LicenseIterator {
 it := &LicenseIterator{}
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {

```

```

resp := &computepb.LicensesListResponse{}
if pageToken != "" {
 req.PageToken = proto.String(pageToken)
}
if pageSize > math.MaxInt32
{
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
} else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

params := url.Values{}
if req != nil && req.Filter != nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from
client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

```

```

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
if err != nil {
 return err
}
if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

```

```

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken,
 nil
}

```

```

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

```

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }
}

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {

```

```

 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

// Build HTTP headers from client and context metadata.
hds
:= []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project", url.QueryEscape(req.GetProject()),
"resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if
err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is

intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licensesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params",
 fmt.Sprintf("%s=%v&%s=%v", "project", url.QueryEscape(req.GetProject()), "resource",
url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err !=
 nil {
 return err
 }
 })
```

```

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3

```

```
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GranteeType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GranteeType = 3
)

// Enum value maps for Permission_GranteeType.
var (
 Permission_GranteeType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2: "GROUP",
 3: "EVERYONE",
 }
 Permission_GranteeType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 }
)
```



```

 "GROUP": 2,
 "EVERYONE": 3,
}
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.

```

```

var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,

```

```

// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user of group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.
 EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
 json:"email_address,omitempty"`
 // Required. The role granted by this permission.
 Role *Permission_Role
 `protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission_Role,oneof"
 json:"role,omitempty"`
}

func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

```

```

}

var
File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,
0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,
0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,
0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,
0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,
0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,
0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,
0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,
0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
0x01, 0x01, 0x12,
0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,
0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,
0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,
0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,
0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,
0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,
0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,
0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,
0x54,
0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,
0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,
0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,
0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,
0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,
0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,
0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,
0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,

```

```

0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69,
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f,
0x72, 0x6f, 0x6c, 0x65, 0x42,
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc
)

```

```

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GrantType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}

```

```

}
var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1beta.Permission.GranteeType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
 google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }
func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
 if File_google_ai_generativelanguage_v1beta_permission_proto != nil {
 return
 }
 file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
 }.Build()
 File_google_ai_generativelanguage_v1beta_permission_proto = out.File
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
 file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
 = nil
 file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0

```

```
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()
}
```



```

req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request
 struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseCodeRequest.
}
resp, err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseCodeRequest.
 }
 resp,
 err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software

```

```

// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"

 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport
 "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newLicenseManagementClientHook clientHook

// LicenseManagementCallOptions contains the retry settings for each method of LicenseManagementClient.
type LicenseManagementCallOptions struct {
 GetLicensePool []gax.CallOption
 UpdateLicensePool []gax.CallOption
 Assign []gax.CallOption
 Unassign []gax.CallOption
 EnumerateLicensedUsers []gax.CallOption
 GetOperation []gax.CallOption
}

func defaultLicenseManagementGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("cloudcommerceconsumerprocurement.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN:443")
 }
}

```

```
,
internaloption.WithDefaultMTLSEndpoint("cloudcommerceconsumerprocurement.mtls.googleapis.com:443"),
internaloption.WithDefaultUniverseDomain("googleapis.com"),
internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
internaloption.WithDefaultScopes(DefaultAuthScopes()),
internaloption.EnableJwtWithScope(),
internaloption.EnableNewAuthLibrary(),
option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
}
}
```

```
func
defaultLicenseManagementCallOptions() *LicenseManagementCallOptions {
return &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
}
}
```

```
func defaultLicenseManagementRESTCallOptions() *LicenseManagementCallOptions {
return
&LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
}
}
```

```
// internalLicenseManagementClient is an interface that defines the methods available from Cloud Commerce
Consumer Procurement API.
type internalLicenseManagementClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 GetLicensePool(context.Context, *procurementpb.GetLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 UpdateLicensePool(context.Context, *procurementpb.UpdateLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 Assign(context.Context, *procurementpb.AssignRequest, ...gax.CallOption) (*procurementpb.AssignResponse,
error)
```

```

Unassign(context.Context, *procurementpb.UnassignRequest,
...gax.CallOption) (*procurementpb.UnassignResponse, error)
EnumerateLicensedUsers(context.Context, *procurementpb.EnumerateLicensedUsersRequest, ...gax.CallOption)
*LicensedUserIterator
GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
}

// LicenseManagementClient is a client for interacting with Cloud Commerce Consumer Procurement API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// Service for managing licenses.
type LicenseManagementClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseManagementClient

 // The call options for this service.
 CallOptions *LicenseManagementCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseManagementClient)
 Close() error {
 return c.internalClient.Close()
 }

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseManagementClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseManagementClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// GetLicensePool gets the license pool.
func (c *LicenseManagementClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {

```

```

return c.internalClient.GetLicensePool(ctx, req, opts...)
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *LicenseManagementClient) UpdateLicensePool(ctx context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 return c.internalClient.UpdateLicensePool(ctx, req, opts...)
}

// Assign assigns a license to a user.
func (c *LicenseManagementClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption) (*procurementpb.AssignResponse, error) {
 return c.internalClient.Assign(ctx, req, opts...)
}

// Unassign unassigns a license from a user.
func (c *LicenseManagementClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest, opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 return c.internalClient.Unassign(ctx, req, opts...)
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *LicenseManagementClient) EnumerateLicensedUsers(ctx context.Context, req *procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 return c.internalClient.EnumerateLicensedUsers(ctx, req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *LicenseManagementClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// licenseManagementGRPCClient is a client for interacting with Cloud Commerce Consumer Procurement API over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.
type licenseManagementGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 // The gRPC API client.

```

```

licenseManagementClient procurementpb.LicenseManagementServiceClient

operationsClient longrunningpb.OperationsClient

//
// The x-goog-* metadata to be sent with each request.
xGoogHeaders []string

logger *slog.Logger
}

// NewLicenseManagementClient creates a new license management service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing licenses.
func NewLicenseManagementClient(ctx context.Context, opts ...option.ClientOption) (*LicenseManagementClient,
error) {
 clientOpts := defaultLicenseManagementGRPCClientOptions()
 if newLicenseManagementClientHook != nil {
 hookOpts, err := newLicenseManagementClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }
 client := LicenseManagementClient{CallOptions: defaultLicenseManagementCallOptions()}

 c := &licenseManagementGRPCClient{
 connPool:
 connPool,
 licenseManagementClient: procurementpb.NewLicenseManagementServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
 }
 c.setGoogleClientInfo()

 client.internalClient = c

 return &client, nil
}

// Connection returns a connection to the API service.

```

```

//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *licenseManagementGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gpic", getVersionClient(), "gax", gax.Version, "grpc",
 grpc.Version)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 logger *slog.Logger
}

// NewLicenseManagementRESTClient creates a new license management service rest client.
//
// Service for managing licenses.
func NewLicenseManagementRESTClient(ctx context.Context, opts ...option.ClientOption)

```

```

(*LicenseManagementClient, error) {
 clientOpts := append(defaultLicenseManagementRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseManagementRESTCallOptions()
 c := &licenseManagementRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseManagementClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseManagementRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://cloudcommerceconsumerprocurement.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://cloudcommerceconsumerprocurement.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN")
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.

```



```

c.httpClient = nil
return nil
}

// Connection
returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseManagementRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *licenseManagementGRPCClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.GetLicensePool, req, settings.GRPC, c.logger,
"GetLicensePool")
 return err
 }, opts...)
 if err != nil {
 return
 nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.UpdateLicensePool, req, settings.GRPC, c.logger,

```

```
"UpdateLicensePool")
```

```
 return err
```

```
}, opts...)
```

```
if err != nil {
```

```
 return nil, err
```

```
}
```

```
return resp, nil
```

```
}
```

```
func (c *licenseManagementGRPCClient) Assign(ctx context.Context,
```

```
req *procurementpb.AssignRequest, opts ...gax.CallOption) (*procurementpb.AssignResponse, error) {
```

```
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}
```

```
hds = append(c.xGoogHeaders, hds...)
```

```
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
```

```
opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
```

```
var resp *procurementpb.AssignResponse
```

```
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
```

```
 var err error
```

```
 resp, err = executeRPC(ctx, c.licenseManagementClient.Assign, req, settings.GRPC, c.logger, "Assign")
```

```
 return err
```

```
}, opts...)
```

```
if err != nil {
```

```
 return nil, err
```

```
}
```

```
return resp, nil
```

```
}
```

```
func (c *licenseManagementGRPCClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
```

```
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
```

```
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}
```

```
hds
```

```
= append(c.xGoogHeaders, hds...)
```

```
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
```

```
opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
```

```
var resp *procurementpb.UnassignResponse
```

```
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
```

```
 var err error
```

```
 resp, err = executeRPC(ctx, c.licenseManagementClient.Unassign, req, settings.GRPC, c.logger, "Unassign")
```

```
 return err
```

```
}, opts...)
```

```
if err != nil {
```

```
 return nil, err
```

```
}
```

```
return resp, nil
```

```
}
```

```

func (c *licenseManagementGRPCClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).EnumerateLicensedUsers[0:len((*c.CallOptions).EnumerateLicensedUsers):len((*c.CallOp
tions).EnumerateLicensedUsers)],
opts...)
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
resp := &procurementpb.EnumerateLicensedUsersResponse{}
if pageToken != "" {
req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
req.PageSize = math.MaxInt32
} else if pageSize != 0 {
req.PageSize = int32(pageSize)
}
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.licenseManagementClient.EnumerateLicensedUsers, req, settings.GRPC, c.logger,
"EnumerateLicensedUsers")
return err
}, opts...)
if err != nil {
return nil, "", err
}

it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int,
pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

```

```

return it
}

func (c *licenseManagementGRPCClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
error {
var err error
resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

// GetLicensePool gets the license pool.
func (c *licenseManagementRESTClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP

```

```

ool)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *licenseManagementRESTClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest,
opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetLicensePool()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetLicensePool().GetName())

```

```

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 params.Add("updateMask", string(field[1:len(field)-1]))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers
:= gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).UpdateLicensePool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)

```

```

if e != nil {
 return nil, e
}
return resp, nil
}

// Assign assigns a license to a user.
func (c *licenseManagementRESTClient) Assign(ctx
context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption) (*procurementpb.AssignResponse,
error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:assign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp
:= &procurementpb.AssignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

```

```

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Assign")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Unassign unassigns a license from a user.
func (c *licenseManagementRESTClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return
 nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:unassign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &procurementpb.UnassignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```



```

if settings.Path != "" {
 baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(),
bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Unassign")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *licenseManagementRESTClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 it := &LicensedUserIterator{}
 req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
 unmarshalOptions := &protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if
 pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err

```

```

}
baseUrl.Path += fmt.Sprintf("/v1/%v:enumerateLicensedUsers", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "EnumerateLicensedUsers")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {

```

```

 return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c
*licenseManagementRESTClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }

```

```

httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
)

func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()

```

```

// This snippet has been automatically
generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
}
for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3
// source: google/ai/generativelanguage/v1alpha/permission.proto

package generativelanguagepb

```

```

import (
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
)

const
(
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GrantType int32

const (
 // The default value. This value is unused.
 Permission_GRANT_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANT_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANT_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

```

```

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
 []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 }

```

```

2: "WRITER",
3: "READER",
}
Permission_Role_value = map[string]int32{
"ROLE_UNSPECIFIED": 0,
"OWNER": 1,
"WRITER": 2,
"READER": 3,
}
)

func (x Permission_Role) Enum() *Permission_Role {
p := new(Permission_Role)
*p = x
return p
}

func (x Permission_Role) String() string
{
return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protorelect.EnumDescriptor {
return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protorelect.EnumType {
return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protorelect.EnumNumber {
return protorelect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service
// accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//

```



```

// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user of group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.
 EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
 json:"email_address,omitempty"`
 // Required. The role granted by this permission.
 Role *Permission_Role
 `protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
 json:"role,omitempty"`
}

func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

```

```

func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,
 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,
 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,
 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,
 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,
 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,
 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,
 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,
 0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,
 0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,
 0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,
 0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,
 0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,
 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,
 0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,
 0x4e,
 0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,
 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,
 0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,
 0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,
 0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,
 0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,
 0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,
 0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,
 0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,
 0x2e, 0x63, 0x6f, 0x6d,
 0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,
 0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,

```

```

0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,
0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

```

```

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes =
 []any{
 (Permission_GrantType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
 }
var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->

```

```

google.ai.generativelanguage.v1alpha.Permission.GrantType
1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
google.ai.generativelanguage.v1alpha.Permission.Role
2, // [2:2] is the sub-list for method output_type
2, // [2:2] is the sub-list for method input_type
2, // [2:2] is the sub-list for extension type_name
2, // [2:2] is the sub-list for extension extendee
0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}
func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
return
}
file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
NumEnums: 2,
NumMessages: 1,
NumExtensions: 0,
NumServices: 0,
},
GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
= nil
file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}

// Copyright 2023 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software

```

```

// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 context "context"
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
 _ "google.golang.org/grpc/status"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)

```

)

// Assignment protocol for a license pool.

```
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
```

// The type of assignment protocol.

//

// Types that are assignable to AssignmentType:

//

// \*AssignmentProtocol\_ManualAssignmentType\_

// \*AssignmentProtocol\_AutoAssignmentType\_

AssignmentType

isAssignmentProtocol\_AssignmentType `protobuf\_oneof:"assignment\_type"``

}

```
func (x *AssignmentProtocol) Reset() {
```

```
 *x = AssignmentProtocol{}
```

```
 mi :=
```

```
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
```

```
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
```

```
 ms.StoreMessageInfo(mi)
```

```
}
```

```
func (x *AssignmentProtocol) String() string {
```

```
 return protoimpl.X.MessageStringOf(x)
```

```
}
```

```
func (*AssignmentProtocol) ProtoMessage() {}
```

```
func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
```

```
 mi :=
```

```
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
```

```
 if x != nil {
```

```
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
```

```
 if ms.LoadMessageInfo() == nil {
```

```
 ms.StoreMessageInfo(mi)
```

```
 }
```

```
 return ms
```

```
 }
```

```
 return mi.MessageOf(x)
```

```
}
```

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.

```
func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
```

```
 return
```

```

file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType {
 if m != nil {
 return m.AssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
 return x.ManualAssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
 return x.AutoAssignmentType
 }
 return nil
}

type isAssignmentProtocol_AssignmentType interface {
 isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_ struct {
 // Allow manual assignments triggered by administrative operations only.
 ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
 `protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type
 AssignmentProtocol_AutoAssignmentType_ struct {
 // Allow automatic assignments triggered by data plane operations.
 AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
 `protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
 }

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {

```



```

state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Identifier. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Required. Assignment protocol for the license pool.
LicenseAssignmentProtocol *AssignmentProtocol
`protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
json:"license_assignment_protocol,omitempty"`
// Output only. Licenses count that are available to be assigned.
AvailableLicenseCount int32
`protobuf:"varint,3,opt,name=available_license_count,json=availableLicenseCount,proto3"
json:"available_license_count,omitempty"`
// Output only. Total number of licenses in the pool.
TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total_license_count,json=totalLicenseCount,proto3"
json:"total_license_count,omitempty"`
}

func (x *LicensePool) Reset() {
 *x = LicensePool{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensePool) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensePool) ProtoMessage() {}

func (x *LicensePool) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.
func (*LicensePool) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{1}
}

func (x *LicensePool) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a
// license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

func (x *GetLicensePoolRequest) Reset() {

```

```

*x = GetLicensePoolRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if
ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The license pool to update.

```

```

//
// The license pool's name field is used to identify the license pool to
// update. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`.
LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
json:"license_pool,omitempty"`
//
Required. The list of fields to update.
UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
json:"update_mask,omitempty"`
}

func (x *UpdateLicensePoolRequest) Reset() {
*x = UpdateLicensePoolRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UpdateLicensePoolRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*UpdateLicensePoolRequest) ProtoMessage() {}

func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor
instead.
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{3}
}

func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {

```

```

if x != nil {
 return x.LicensePool
}
return nil
}

func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

// Request message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.prourement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func
(x *AssignRequest) Reset() {
 *x = AssignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[4]
 if x != nil {

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{4}
}

func (x *AssignRequest) GetParent()
string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

```

```

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x
 != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{5}
}

// Request message for
//
// [LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset()
{
 *x = UnassignRequest{}
 mi :=

```

```

&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{6}
}

func (x *UnassignRequest) GetParent() string {
 if x != nil {
 return
 x.Parent
 }
 return ""
}

func (x *UnassignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//

```



```

[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {

```

```

state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Required. License pool name.
Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
// Optional. The maximum number of users to return. The service may return
// fewer than this value.
PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3"
json:"page_size,omitempty"`
// Optional. A page token, received from a previous `EnumerateLicensedUsers`
// call. Provide this to retrieve the subsequent page.
PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

func (x *EnumerateLicensedUsersRequest) Reset() {
*x = EnumerateLicensedUsersRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo()
== nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{8}
}

```

```

}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

// A licensed user.
type LicensedUser struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Username.
 // Format: `name@domain.com`.
 Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
 //
 // Output only. Timestamp when the license was assigned.
 AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3"
 json:"assign_time,omitempty"`
 // Output only. Timestamp when the license was recently used. This may not be
 // the most recent usage time, and will be updated regularly (within 24
 // hours).
 RecentUsageTime *timestamppb.Timestamp
 `protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3"
 json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {
 *x = LicensedUser{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if
x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }

```

```

}
return nil
}

// Response message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.prourement.v1.License
ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The list of licensed users.
 LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
 json:"licensed_users,omitempty"`
 // A token that can be sent as `page_token` to retrieve the next page.
 // If this field is omitted, there are no subsequent pages.
 NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
 json:"next_page_token,omitempty"`
}

func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersResponse) ProtoMessage() {}

func
(x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[10]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
}

```

```

return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{10}
}

func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
 return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=

```

```

&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated:
// Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Optional. The time to live for an inactive license. After this time has
 // passed, the license will be automatically unassigned from the user. Must
 // be at least 7 days, if set. If unset, the license will never expire.
 InactiveLicenseTtl *durationpb.Duration
 `protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
 json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {

```

```

mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
}
return mi.MessageOf(x)
}

```

```

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

```

```

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration {
 if x != nil {
 return x.InactiveLicenseTtl
 }
 return nil
}

```

```

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protoreflect.FileDescriptor

```

```

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{
 0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
 0x6f,
 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
 0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
 0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,
 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,
 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,

```



0x62, 0x75, 0x66, 0x2f, 0x64,  
0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,  
0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,  
0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73,  
0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,  
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,  
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01,  
0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42, 0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42, 0x03, 0xe0,

0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,

0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,  
0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69,  
0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,  
0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc5, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,

0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x8b, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x71, 0xda, 0x41, 0x18, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x50, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x40, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,  
0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x2f, 0x2a, 0x7d, 0x12, 0xdf, 0x01,  
0x0a, 0x06, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63,  
0x65,  
0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72,  
0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73,  
0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f,  
0x3a, 0x01, 0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73,  
0x2f, 0x2a, 0x2f, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12,  
0xe7, 0x01, 0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61,

0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x5a, 0xda,  
0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67,  
0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72,  
0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d,  
0x3a, 0x75, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e,  
0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55,  
0x73, 0x65, 0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,  
0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e,  
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73,  
0x65, 0x22, 0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93,  
0x02, 0x4c, 0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d,  
0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f,  
0x2a, 0x2f, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74,  
0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63,  
0xca, 0x41, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63,  
0x6f, 0x6d, 0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f,  
0x61, 0x75, 0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66,  
0x6f, 0x72, 0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,  
0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63,  
0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75,  
0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76,  
0x69, 0x63, 0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f,  
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70,  
0x69, 0x76, 0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70,  
0x62, 0x3b, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa,  
0x02, 0x2d, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x43,  
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,

```

0x2e, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca,
0x02, 0x2d, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
0x5c, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea,
0x02, 0x32, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a,
0x3a, 0x43, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75,
0x6d, 0x65, 0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74,
0x3a, 0x3a, 0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
= file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f
unc() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
 })
 return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
 (*AssignmentProtocol)(nil),
 // 0: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 (*LicensePool)(nil),
 // 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
 (*GetLicensePoolRequest)(nil),
 // 2:
 google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 (*UpdateLicensePoolRequest)(nil),
 // 3:
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 (*AssignRequest)(nil),
 // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
 (*AssignResponse)(nil),
 // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
 (*UnassignRequest)(nil),
 // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 (*UnassignResponse)(nil),
 // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 (*EnumerateLicensedUsersRequest)(nil),
 // 8:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 (*LicensedUser)(nil),

```

```

 // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
(*EnumerateLicensedUsersResponse)(nil), // 10:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
(*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
(*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
(*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask
(*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
(*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =
[]int32{
 11, // 0:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name
-> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12, // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name -
> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
 13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
google.protobuf.FieldMask
 14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
google.protobuf.Timestamp
 14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name
-> google.protobuf.Timestamp
 9, // 7:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensedUser
 15, // 8:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
ype_name -> google.protobuf.Duration
 2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
> google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 3, // 10:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
google.cloud.commerce.consumer.procurement.v1.AssignRequest
 6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type
-> google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 8, // 13:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_typ
e -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 1, // 14:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->

```

```

google.cloud.commerce.consumer.procurement.v1.LicensePool
1, // 15:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
google.cloud.commerce.consumer.procurement.v1.AssignResponse
7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
google.cloud.commerce.consumer.procurement.v1.UnassignResponse
10, // 18:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_ty
pe
-> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14, // [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}
func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File:
 protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:

```



```

file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
}.Build()
File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs
= nil
}

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// LicenseManagementServiceClient is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {
 // Gets the license pool.
 GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool,
error)
 // Assigns a license to a user.
 Assign(ctx
context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption)
(*EnumerateLicensedUsersResponse, error)
}

type licenseManagementServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
 return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts
...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)

```

```

err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool", in, out, opts...)
if err != nil {
 return nil, err
}
return
out, nil
}

func (c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in
*UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption)
(*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign", in,
out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers", in, out,

```

```

opts...)
if err != nil {
 return nil, err
}
return out, nil
}

// LicenseManagementServiceServer is the server API for LicenseManagementService service.
type LicenseManagementServiceServer interface {
 // Gets the license pool.
 GetLicensePool(context.Context, *GetLicensePoolRequest) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(context.Context, *UpdateLicensePoolRequest) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(context.Context, *AssignRequest)
 (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(context.Context, *UnassignRequest) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(context.Context, *EnumerateLicensedUsersRequest)
 (*EnumerateLicensedUsersResponse, error)
}

// UnimplementedLicenseManagementServiceServer can be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer struct {
}

func (*UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)
(*UnassignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest) (*EnumerateLicensedUsersResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

```

```

func RegisterLicenseManagementServiceServer(s *grpc.Server, srv LicenseManagementServiceServer) {
 s.RegisterService(&_LicenseManagementService_serviceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(GetLicensePoolRequest)
 if err := dec(in);
 err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/GetLicensePool",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(UpdateLicensePoolRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/UpdateLicensePool",
 }
 handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(AssignRequest)

```

```

if err := dec(in); err != nil {
 return nil, err
}
if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Assign(ctx, in)
}
info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/Assign",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv
interface{}, ctx context.Context, dec func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
 in := new(UnassignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/Unassign",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(EnumerateLicensedUsersRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if
 interceptor == nil {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,

```

```

FullMethod:
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
}
return interceptor(ctx, in, info, handler)
}

var _LicenseManagementService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.commerce.consumer.procurement.v1.LicenseManagementService",
 HandlerType: (*LicenseManagementServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "GetLicensePool",
 Handler: _LicenseManagementService_GetLicensePool_Handler,
 },
 {
 MethodName: "UpdateLicensePool",
 Handler: _LicenseManagementService_UpdateLicensePool_Handler,
 },
 {
 MethodName: "Assign",
 Handler: _LicenseManagementService_Assign_Handler,
 },
 {
 MethodName: "Unassign",
 Handler: _LicenseManagementService_Unassign_Handler,
 },
 {
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works



that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

package compute

import (

"bytes"

"context"

"fmt"

"log/slog"

"net/http"

"net/url"

"time"

computepb "cloud.google.com/go/compute/apiv1/computepb"

gax "github.com/googleapis/gax-go/v2"

"google.golang.org/api/option"

"google.golang.org/api/option/internaloption"

httptransport "google.golang.org/api/transport/http"

"google.golang.org/grpc"

"google.golang.org/protobuf/encoding/protojson"

)

var

newLicenseCodesClientHook clientHook

```

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 Close()
 error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

```

```

}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return c.internalClient.Close()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

```

```

// The http client.
httpClient *http.Client

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing LicenseCodesClient
CallOptions **LicenseCodesCallOptions

logger *slog.Logger
}

// NewLicenseCodesRESTClient
// creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
 }
}

```

```

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN")
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return nil
}

// Connection
// returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)

```

```

opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.LicenseCode{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified
// resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace
// images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }

```



```

baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v/testIamPermissions",
req.GetProject(), req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type",
"application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

```

## 1.87 go-jose-go-jose 4.1.0

## 1.87.1 Available under license :

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.88 confluentinc-confluent-kafka-go 2.10.0

## 1.88.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)
```

```
// Lists received licenses.
```

```
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
 if params == nil {
 params = &ListReceivedLicensesInput{
 }
}
```

```
 result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
 if err != nil {
 return nil, err
 }
```

```
 out := result.(*ListReceivedLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}
```

```
type ListReceivedLicensesInput struct {
```

```
 // Filters to scope the results. The following filters are supported:
```

```
 //
```

```
 // - ProductSKU
```

```
 //
```

```
 // - Status
```

```
 //
```

```
 //
```

```
 // - Fingerprint
```

```
 //
```

```
 // - IssuerName
```

```

//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```



```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err =
addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense  
Initialize stack step  
spanInitializeStart  
RegisterServiceMetadata  
legacyEndpointContextSetter  
SetLogger  
OperationInputValidation  
spanInitializeEnd  
Serialize stack step  
spanBuildRequestStart  
setOperationInput  
ResolveEndpoint  
OperationSerializer  
Build stack step  
ClientRequestID  
ComputeContentLength  
SetCredentialSourceMiddleware  
UserAgent  
AddTimeOffsetMiddleware  
RecursionDetection  
spanBuildRequestEnd  
Finalize stack step  
spanRetryLoop  
Retry  
RetryMetricsHeader  
ResolveAuthScheme  
GetIdentity  
ResolveEndpointV2  
disableHTTPS  
ComputePayloadHash  
setLegacyContextSigningOptions  
Signing  
Deserialize stack step  
AddRawResponseToMetadata  
ErrorCloseResponseBody  
CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger  
name: License Scan  
  
on: [pull\_request]  
  
jobs:  
 licensescan:

```

name: License Scan
runs-on: ubuntu-latest
strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: ${ github.base_ref }
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
 ${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit
 path: scancode-toolkit
 fetch-depth: 1
 - name: Set up Python ${ matrix.python-version }
 uses: actions/setup-python@v2
 with:
 python-version: ${ matrix.python-version }
 # ScanCode
 - name: Self-configure scancode
 working-directory: ./scancode-toolkit
 run: ./scancode --help
 - name: Run Scan code on target
 run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
 toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
 - name: Run Scan code on pr ref
 run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
 toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
 # compare
 - name: License test
 run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
 are the same. Success."; exit 0; fi
ListLicenses
Initialize stack step

```

```

spanInitializeStart
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
Retry
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
 if params == nil {
 params = &CreateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CreateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CreateLicenseInput struct {

 // License beneficiary.
 //
 // This member is required.
 Beneficiary *string

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 //
 // This member is required.
 ClientToken *string

 // Configuration for consumption of the license. Choose a provisional
 // configuration for workloads running with continuous connectivity. Choose a
 // borrow configuration for workloads with offline usage.
 //
 // This member is required.
 ConsumptionConfiguration *types.ConsumptionConfiguration

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.Entitlement

```

```

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}, middleware.After)
 if err != nil {
 return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if
 err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
 if err = addClientRequestID(stack); err != nil {
 return err
 }
 if err = addComputeContentLength(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {

```



```

 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err
= addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}

```

```

}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err
= addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "CreateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/grafana/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
 if params == nil {
 params = &DisassociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
 if err != nil {
 return nil, err
 }
}

```

```

}

out := result.(*DisassociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

 // The type of license to remove from the workspace.
 //
 // This member is required.
 LicenseType types.LicenseType

 //
 // The ID of the workspace to remove the Grafana Enterprise license from.
 //
 // This member is required.
 WorkspaceId *string

 noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

 // A structure containing information about the workspace.
 //
 // This member is required.
 Workspace *types.WorkspaceDescription

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
 if err != nil {
 return err
 }

```

```

}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
 if params == nil {
 params = &DeleteLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*DeleteLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type DeleteLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

 // Current version of the license.
 //

```

```

// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

 // Date when the license is deleted.
 DeletionDate *string

 // License status.
 Status types.LicenseDeletionStatus

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
 if err != nil {
 return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addLegacyEndpointContextSetter(stack, options); err !=
 nil {
 return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
 if err = addClientRequestID(stack); err != nil {
 return err
 }
 if err = addComputeContentLength(stack); err != nil {

```

```

 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
 return err
}

```



```

}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "DeleteLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
 if params == nil {
 params = &ListLicensesInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*ListLicensesOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type ListLicensesInput struct {

 // Filters to scope the results. The following filters are supported:
 //
 // - Beneficiary
 //
 // - ProductSKU
 //
 // - Fingerprint
 //
 // - Status
 Filters
 []types.Filter

 // Amazon Resource Names (ARNs) of the licenses.
 LicenseArns []string

 // Maximum number of results to return in a single call.
 MaxResults *int32

 // Token for the next set of results.
 NextToken *string

 noSmithyDocumentSerde
}

```

```

type ListLicensesOutput struct {

 // License details.
 Licenses []types.License

 // Token for the next set of results.
 NextToken *string

 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
 if err != nil {
 return err
 }
 if err := addProtocolFinalizerMiddlewares(stack,
 options, "ListLicenses"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
 if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
 }
 if err = addClientRequestID(stack); err != nil {
 return err
 }
 if err = addComputeContentLength(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
}

```

```

}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
}

```

```

if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack);
err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "ListLicenses",
 }
}

DisassociateLicense
Initialize stack step
spanInitializeStart
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```

RecursionDetection  
spanBuildRequestEnd  
Finalize stack step  
spanRetryLoop  
Retry  
RetryMetricsHeader  
ResolveAuthScheme  
GetIdentity  
ResolveEndpointV2  
disableHTTPS  
ComputePayloadHash  
setLegacyContextSigningOptions  
Signing  
Deserialize stack step  
AddRawResponseToMetadata  
ErrorCloseResponseBody  
CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger  
CheckInLicense  
Initialize stack step  
spanInitializeStart  
RegisterServiceMetadata  
legacyEndpointContextSetter  
SetLogger  
OperationInputValidation  
spanInitializeEnd  
Serialize stack step  
spanBuildRequestStart  
setOperationInput  
ResolveEndpoint  
OperationSerializer  
Build stack step  
ClientRequestID  
ComputeContentLength  
SetCredentialSourceMiddleware  
UserAgent  
AddTimeOffsetMiddleware  
RecursionDetection  
spanBuildRequestEnd  
Finalize stack step  
spanRetryLoop  
Retry  
RetryMetricsHeader

ResolveAuthScheme  
GetIdentity  
ResolveEndpointV2  
disableHTTPS  
ComputePayloadHash  
setLegacyContextSigningOptions  
Signing  
Deserialize stack step  
AddRawResponseToMetadata  
ErrorCloseResponseBody  
CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger  
CheckoutBorrowLicense  
Initialize stack step  
spanInitializeStart  
RegisterServiceMetadata  
legacyEndpointContextSetter  
SetLogger  
OperationInputValidation  
spanInitializeEnd  
Serialize stack step  
spanBuildRequestStart  
setOperationInput  
ResolveEndpoint  
OperationSerializer  
Build stack step  
ClientRequestID  
ComputeContentLength  
SetCredentialSourceMiddleware  
UserAgent  
AddTimeOffsetMiddleware  
RecursionDetection  
spanBuildRequestEnd  
Finalize stack step  
spanRetryLoop  
Retry  
RetryMetricsHeader  
ResolveAuthScheme  
GetIdentity  
ResolveEndpointV2  
disableHTTPS  
ComputePayloadHash  
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
 if params == nil {
 params = &GetLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*GetLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type GetLicenseInput struct {

 // Amazon Resource Name (ARN) of the license.
 //
 // This member is required.
 LicenseArn *string

```



```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return
nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "GetLicense",
 }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

spanInitializeStart

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

```

Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
Retry
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.

```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
 if params == nil {
 params = &CheckoutBorrowLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CheckoutBorrowLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutBorrowLicenseInput struct {

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 //
 the request.
 //
 // This member is required.
 ClientToken *string

 // Digital signature method. The possible value is JSON Web Signature (JWS)
 // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
 //
 // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
 //
 // This member is required.
 DigitalSignatureMethod types.DigitalSignatureMethod

 // License entitlements. Partial checkouts are not supported.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Amazon Resource Name (ARN) of the license. The license must use the borrow
 // consumption configuration.
 //
 // This member is required.
 LicenseArn *string

 // Information about constraints.
 CheckoutMetadata []types.Metadata

```

```

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
 return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
 return err
}
if err = addClientRequestID(stack); err != nil {
 return err
}
if err = addComputeContentLength(stack); err != nil {
 return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
}

```



```

return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "CheckoutBorrowLicense",
}
}
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
 "smithy.api#http": {
 "code": 200,

```

```

 "method": "POST",
 "uri": "/subscription/DeregisterSubscriptionProvider"
 },
 "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
 "type": "structure",
 "members": {
 "SubscriptionProviderArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The type of name to filter by.</p>"
 }
 },
 "Values": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
 "traits": {
 "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
 }
 }
 },
 "Operator": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
 "traits": {
 "smithy.api#documentation": "<p>An operator for filtering results.</p>"
 }
 }
}

```

```

 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n Filters can be used to match a set of resources by specific criteria.</p>"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
 "type":
"list",
 "member": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
 },
 "output": {
 "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/subscription/GetRegisteredSubscriptionProvider"
 },
 "smithy.api#idempotent": {}
 }
 }
 }
}

```

```

 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
 "type": "structure",
 "members": {
 "SubscriptionProviderArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
 "traits": {
 "smithy.api#documentation":
 "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
 "type": "structure",
 "members": {
 "SubscriptionProviderArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
 }
 },
 "SubscriptionProviderSource": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
 "traits": {
 "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
 }
 },
 "SecretArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
 }
 },
 "SubscriptionProviderStatus": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
 "traits": {
 "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
 }
 }
 }
 }
 }
}

```



```

 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
 "type": "structure",
 "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
 "type": "structure",
 "members": {
 "LinuxSubscriptionsDiscovery": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
 "traits": {
 "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
 }
 },
 "LinuxSubscriptionsDiscoverySettings": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
 "traits": {
 "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
 "traits": {
 "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
 }
 },
 "StatusMessage": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
 "traits": {
 "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
 }
 },
 "HomeRegions": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
 "traits": {
 "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n subscriptions.</p>"
 }
 }
 }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
 "type": "structure",
 "members": {
 "AmiId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
 }
 },
 "InstanceID": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
 }
 },
 "InstanceType": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The instance type of the resource.</p>"
 }
 },
 "AccountID": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
 }
 },
 "Status": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The status of the instance.</p>"
 }
 },
 "Region": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
 }
 },
 "UsageOperation": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n operation values in the <i>License Manager User
Guide</i>.</p>"
 }
 }
 }
}

```

```

 }
 },
 "ProductCode": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
 "traits": {
 "smithy.api#documentation": "<p>The product code for the instance. For more information, see Usage operation values in the <i>License Manager User Guide</i> .</p>"
 }
 },
 "LastUpdatedTime": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
 }
 },
 "SubscriptionName": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
 }
 },
 "OsVersion": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
 }
 },
 "SubscriptionProviderCreateTime": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
 }
 },
 "SubscriptionProviderUpdateTime": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
 }
 },
 "DualSubscription": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```



\\n\\t\\tthe same software on your instance.</p>"

```
 }
 },
 "RegisteredWithSubscriptionProvider": {
 "target": "smithy.api#String",
 "traits": {

 "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \\n\\t\\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
 "smithy.api#error": "server"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
 "type": "service",
 "version": "2018-05-10",
 "operations": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```

 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
 }
],
 "traits": {
 "aws.api#service": {
 "sdkId": "License Manager Linux Subscriptions",
 "arnNamespace": "license-manager-linux-subscriptions",
 "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
 },
 "aws.auth#sigv4": {
 "name": "license-manager-linux-subscriptions"
 },
 "aws.protocols#restJson1": {},
 "smithy.api#cors": {
 "additionalAllowedHeaders": [
 "*",
 "content-type",
 "x-amz-content-sha256",
 "x-amz-user-agent",
 "x-amzn-platform-id",
 "x-amzn-trace-id"
],
 "additionalExposedHeaders": [
 "x-amzn-errortype",
 "x-amzn-requestid",

```

```

 "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
 "version": "1.0",
 "parameters": {
 "Region": {
 "builtIn": "AWS::Region",
 "required": false,
 "documentation": "The AWS region used to dispatch the request.",
 "type": "String"
 },
 "UseDualStack": {
 "builtIn": "AWS::UseDualStack",
 "required": true,
 "default": false,
 "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
 "type": "Boolean"
 },
 "UseFIPS": {
 "builtIn": "AWS::UseFIPS",
 "required": true,
 "default": false,
 "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
 "type": "Boolean"
 },
 "Endpoint": {
 "builtIn": "SDK::Endpoint",
 "required": false,
 "documentation": "Override the endpoint used to send this request",
 "type": "String"
 }
 },
 "rules": [
 {
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Endpoint"
 }
]
 }
]
 }
]
}

```

```

 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
]
 }
],
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
 "type": "error"
},
{
 "conditions": [],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
]
 }
]
},
 "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
 "type": "error"
},
{
 "conditions": [],
 "endpoint": {
 "url": {
 "ref": "Endpoint"
 },
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
}

```

```

 }
],
 "type": "tree"
}
],
"type": "tree"
},
{
 "conditions": [],
 "rules": [
 {
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Region"
 }
]
 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "aws.partition",
 "argv": [
 {
 "ref": "Region"
 }
],
 "assign": "PartitionResult"
 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
]
 }
]
 }
]
 }
]
}

```

```

 },
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
],

 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 }
]
 }
]
 },
 {
 "fn": "supportsFIPS"
 }
]
 },
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsDualStack"
]
 }
]
 }
],

```

```

 "rules": [
 {
 "conditions": [],

"rules": [

 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
],

 "type": "tree"
 },
 "type": "tree"
 },
 {
 "conditions": [],
 "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
 "type": "error"
 },
],
 "type": "tree"
 },
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
],
 "rules": [
 {
 "conditions": [
 {

```

```

 "fn": "booleanEquals",
 "argv": [
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsFIPS"
]
 },
 true
]
 },
 "rules": [
 {
 "conditions": [],
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
],
 "type": "tree"
 }
],
 "type": "tree"
},
{
 "conditions": [],
 "error": "FIPS is enabled
but this partition does not support FIPS",
 "type": "error"
}
],
"type": "tree"
},
{

```



```

 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsDualStack"
]
 }
]
 }
]
 },
 {
 "conditions": [],
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
]
 }
]
 }

```

```

 "type": "tree"
 }

],
 "type": "tree"
},
{
 "conditions": [],
 "error": "DualStack is enabled but this partition does not support DualStack",
 "type": "error"
}
],
"type": "tree"
},
{
 "conditions": [],
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
],
 "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
 "conditions": [],
 "error": "Invalid Configuration: Missing Region",
 "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
 "testCases": [

```

```

{
 "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
 }
 }
}

```

```

 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},

```

```

{
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
 }
 },

```

```

 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack

```

```

enabled",
 "expect": {
 "error": "DualStack is enabled but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS":
false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
 }
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},

```

```

{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "error": "DualStack is enabled but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
 }
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
 "expect": {
 "endpoint": {
 "url": "https://example.com"
 }
 },
},
{
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
},
{
 "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
 "expect": {
 "endpoint": {
 "url": "https://example.com"
 }
 },
},

```



```

 "params": {
 "UseFIPS": false,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
 },
 {
 "documentation": "For custom endpoint with fips enabled and dualstack disabled",
 "expect": {
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
 },
 {
 "documentation": "For custom endpoint with fips disabled and dualstack enabled",
 "expect": {
 "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": true,
 "Endpoint": "https://example.com"
 }
 },
 {
 "documentation": "Missing region",
 "expect": {
 "error": "Invalid Configuration: Missing Region"
 }
 }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {

```

```

 "name": "Enabled",
 "value": "Enabled",
 "documentation": "Enabled LinuxSubscriptionsDiscovery"
 },
 {
 "name": "Disabled",
 "value": "Disabled",
 "documentation": "Disabled LinuxSubscriptionsDiscovery"
 }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
 "type": "structure",
 "members": {
 "SourceRegions": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
 "traits": {
 "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
 "smithy.api#required": {}
 }
 },
 "OrganizationIntegration": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
 "traits": {
 "smithy.api#documentation":
"<p>Details if you have enabled resource discovery across your accounts in Organizations.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux subscriptions.</p>"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Lists the running Amazon EC2 instances that were discovered with commercial Linux\n subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/subscription/ListLinuxSubscriptionInstances"
 },
 "smithy.api#idempotent": {},
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "Instances"
 }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest":
{
 "type": "structure",
 "members": {
 "Filters": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>An array of structures that you can use to filter the results by your specified criteria. \n \tFor example, you can specify <code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto streamline results:</p>\n \n \n <p>\n <code>AccountID</code>\n </p>\n \n \n <p>\n <code>AmiID</code>\n </p>\n \n \n <p>\n <code>DualSubscription</code>\n </p>\n \n \n <p>\n <code>InstanceID</code>\n </p>\n \n \n <p>\n <code>InstanceType</code>\n </p>\n \n \n <p>\n <code>ProductCode</code>\n </p>\n \n \n <p>\n <code>Region</code>\n </p>\n \n \n <p>\n <code>Status</code>\n </p>\n \n \n <p>\n <code>UsageOperation</code>\n </p>\n \n \n <p>For each filter, you can use one of the following <code>Operator</code>"
 }
 }
 }
}

```

```

values to \n \t\tdefine the behavior of the filter:</p>\n \n \n <p>\n <code>contains</code>\n </p>\n \n \n <p>\n <code>equals</code>\n </p>\n \n \n <p>\n <code>Notequal</code>\n </p>\n \n \n }\n },\n "MaxResults": {\n "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",\n "traits": {\n "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"\n }\n },\n "NextToken": {\n "target": "smithy.api#String",\n "traits": {\n "smithy.api#documentation": "<p>A token to specify where to\nstart paginating. This \n\t\tis the nextToken from a previously truncated response.</p>",\n "smithy.api#length": {\n "min": 1,\n "max": 16384\n }\n }\n }\n },\n "traits": {\n "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if\nparameters in request increases."\n }\n },\n "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {\n "type": "structure",\n "members": {\n "Instances": {\n "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",\n "traits": {\n "smithy.api#documentation": "<p>An array that contains instance objects.</p>"\n }\n },\n "NextToken": {\n\n "target": "smithy.api#String",\n "traits": {\n "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\t\tfield\nisn't empty, there are additional elements that the service hasn't \n\t\tincluded in this request. Use this token with the\nnext request to retrieve \n\t\tadditional objects.</p>"\n }\n }\n }\n }\n}

```

```

 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\nOrganizations.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/subscription/ListLinuxSubscriptions"
 },
 "smithy.api#idempotent": {},
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 }
 }
 },
 "items": "Subscriptions"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
 "type": "structure",
 "members": {
 "Filters": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't

```

```

match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n \n
\n <p>\n <code>Subscription</code>\n </p>\n \n \n
<p>The
valid Operators for this filter are:</p>\n \n \n <p>\n <code>contains</code>\n </p>\n \n \n <p>\n <code>equals</code>\n </p>\n \n \n <p>\n <code>NotEqual</code>\n </p>\n \n "
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
 }
 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis
the nextToken from a previously truncated response.</p>",
 "smithy.api#length": {
 "min": 1,
 "max": 16384
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if
parameters in request increases."
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
 "type": "structure",
 "members": {
 "Subscriptions": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
 "traits": {
 "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
 }
 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
 }
 }
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
 "type": "operation",
 "input": {
 "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
 },
 "output": {
 "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>List Bring Your Own License (BYOL) subscription registration
resources for your account.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/subscription/ListRegisteredSubscriptionProviders"
 },
 "smithy.api#idempotent": {},
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "RegisteredSubscriptionProviders"
 }
 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
 "type": "structure",
 "members": {
 "SubscriptionProviderSources": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```







```

 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
 "type": "structure",
 "members": {
 "tags": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "EQUAL",
 "value": "Equal",
 "documentation": "Equal operator"
 },
 {
 "name": "NOT_EQUAL",
 "value": "NotEqual",
 "documentation": "Not equal operator"
 },
 {
 "name": "CONTAINS",
 "value": "Contains",
 "documentation": "Contains operator"
 }
],
 "smithy.api#length": {
 "min": 1,
 "max": 20
 }
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration":
{
 "type": "string",
 "traits": {
 "smithy.api#enum": [

```

```

 {
 "name": "Enabled",
 "value": "Enabled",
 "documentation": "Enabled OrganizationIntegration"
 },
 {
 "name": "Disabled",
 "value": "Disabled",
 "documentation": "Disabled OrganizationIntegration"
 }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
 "type": "list",
 "member": {
 "target": "smithy.api#String"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/subscription/RegisterSubscriptionProvider"
 },
 "smithy.api#idempotent": {}
 }
}

```



```

 }
 },
 "SubscriptionProviderStatus": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
 "traits": {
 "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
 }
 }
},
"traits": {
 "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider":
{
 "type": "structure",
 "members": {
 "SubscriptionProviderArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription
provider resource that you registered.</p>"
 }
 },
 "SubscriptionProviderSource": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
 "traits": {
 "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
 }
 },
 "SecretArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
 }
 },
 "SubscriptionProviderStatus": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
 "traits": {
 "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
 }
 }
 },

```

```

"SubscriptionProviderStatusMessage": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
 }
},
"LastSuccessfulDataRetrievalTime": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
 }
},
"traits": {
 "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList":
{
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
 "smithy.api#error": "client"
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
 }
}

```

```

 },
 "com.amazonaws.licenseanagerlinuxsubscriptions#Status": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "InProgress",
 "value": "InProgress",
 "documentation": "InProgress status"
 },
 {
 "name": "Completed",
 "value": "Completed",
 "documentation": "Completed status"
 },
 {
 "name": "Successful",
 "value": "Successful",
 "documentation": "Successful status"
 },
 {
 "name": "Failed",
 "value": "Failed",
 "documentation": "Failed status"
 }
]
 }
 },
 "com.amazonaws.licenseanagerlinuxsubscriptions#StringList": {
 "type": "list",
 "member": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max": 100
 }
 }
 },
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max": 100
 },
 "smithy.api#uniqueItems": {}
 }
 }
 }

```

```

 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
 "type": "map",
 "key": {
 "target": "smithy.api#String"
 },
 "value": {
 "target": "smithy.api#String"
 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the subscription.</p>"
 }
 },
 "Type": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
 }
 },
 "InstanceCount": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {

```





```

"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList": {
 "type": "list",
 "member": {
 "target": "smithy.api#String"
 },
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 50
 },
 "smithy.api#sensitive": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
 "smithy.api#http": {
 "method": "PUT",
 "uri": "/tags/{resourceArn}"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
 "type": "structure",
 "members": {
 "resourceArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```



```

 "target": "smithy.api#String"
 }
},
"traits": {
 "smithy.api#documentation":
"<p>The request was denied due to request throttling.</p>",
 "smithy.api#error": "client"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
 "smithy.api#http": {
 "method": "DELETE",
 "uri": "/tags/{resourceArn}"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
 "type": "structure",
 "members": {
 "resourceArn": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
 "smithy.api#httpLabel": {},
 "smithy.api#required": {}
 }
 }
 },
 "tagKeys": {

```



```

 "smithy.api#idempotent": {}
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
 "type": "structure",
 "members": {
 "LinuxSubscriptionsDiscovery": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
 "traits": {
 "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
 "smithy.api#required": {}
 }
 },
 "LinuxSubscriptionsDiscoverySettings": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
 "traits": {
 "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n integration has been enabled, and which Regions data will be aggregated from.</p>",
 "smithy.api#required": {}
 }
 },
 "AllowUpdate": {
 "target": "smithy.api#Boolean",
 "traits": {
 "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
 }
 }
 }
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
 "type": "structure",
 "members": {
 "LinuxSubscriptionsDiscovery": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
 "traits": {
 "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
 }
 },
 "LinuxSubscriptionsDiscoverySettings": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
 "traits": {
 "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n integration has been enabled, and which Regions data will be aggregated from.</p>"
 }
 }
 }
 }
}

```

```

 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
 "traits": {
 "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
 }
 },
 "StatusMessage": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
 "traits": {
 "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
 }
 },
 "HomeRegions": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
 "traits": {
 "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n subscriptions.</p>"
 }
 }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
 "smithy.api#error": "client"
 }
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```

```

"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
 if params == nil {
 params = &AssociateLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx,
 "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*AssociateLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type AssociateLicenseInput struct {

 // The type of license to associate with the workspace.
 //
 // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
 // trials.
 //
 // This member is required.
 LicenseType types.LicenseType

 // The ID of the workspace to associate the license with.
 //
 // This member is required.
 WorkspaceId *string

 // A token from Grafana Labs that ties your Amazon Web Services account with a
 // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
 //
 // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-

```



Grafana-Enterprise.html#AMG-workspace-register-enterprise

GrafanaToken \*string

noSmithyDocumentSerde

}

type AssociateLicenseOutput

struct {

// A structure containing data about the workspace.

//

// This member is required.

Workspace \*types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c \*Client) addOperationAssociateLicenseMiddlewares(stack \*middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsRestjson1\_serializeOpAssociateLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsRestjson1\_deserializeOpAssociateLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack,

options); err != nil {

return err

}

if err = addClientRequestID(stack); err != nil {

return err

}

if err = addComputeContentLength(stack); err != nil {

return err

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
}
if err = addComputePayloadSHA256(stack); err != nil {
 return err
}
if err = addRetry(stack, options); err != nil {
 return err
}
if err = addRawResponseToMetadata(stack); err != nil {
 return err
}
if err = addRecordResponseTiming(stack); err != nil {
 return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
 return err
}
if err = addClientUserAgent(stack, options); err != nil {
 return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
 return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
 return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
 return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
 return err
}
if err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
 return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack);
err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "AssociateLicense",
 }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
 if params == nil {
 params = &CheckoutLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
 if err != nil {
 return nil, err
 }

 out := result.(*CheckoutLicenseOutput)
 out.ResultMetadata = metadata
 return out, nil
}

type CheckoutLicenseInput struct {

 //
 // Checkout type.
 //
 // This member is required.
 CheckoutType types.CheckoutType

 // Unique, case-sensitive identifier that you provide to ensure the idempotency of
 // the request.
 //
 // This member is required.
 ClientToken *string

 // License entitlements.
 //
 // This member is required.
 Entitlements []types.EntitlementData

 // Key fingerprint identifying the license.
 //
 // This member is required.
 KeyFingerprint *string

```

```

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
 if err != nil {
 return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addLegacyEndpointContextSetter(stack, options); err != nil {
 return err
 }
 if err = addSetLoggerMiddleware(stack,
options); err != nil {
 return err
 }
 if err = addClientRequestID(stack); err != nil {
 return err
 }
 if err = addComputeContentLength(stack); err != nil {
 return err
 }
 if err = addResolveEndpointMiddleware(stack, options); err != nil {
 return err
 }
 if err = addComputePayloadSHA256(stack); err != nil {
 return err
 }
 if err = addRetry(stack, options); err != nil {
 return err
 }
 if err = addRawResponseToMetadata(stack); err != nil {
 return err
 }
 if err = addRecordResponseTiming(stack); err != nil {
 return err
 }
 if err = addSpanRetryLoop(stack, options); err != nil {
 return err
 }
 if err = addClientUserAgent(stack, options); err != nil {

```

```

return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}

```

```

}
if err = addSpanBuildRequestStart(stack);
err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
 return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "CheckoutLicense",
 }
}

```

```

CreateLicense
Initialize stack step
 spanInitializeStart
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 spanInitializeEnd
Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 OperationSerializer
Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
Finalize stack step
 spanRetryLoop
 Retry
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS

```



ComputePayloadHash  
setLegacyContextSigningOptions  
Signing  
Deserialize stack step  
AddRawResponseToMetadata  
ErrorCloseResponseBody  
CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger  
GetLicense  
Initialize stack step  
spanInitializeStart  
RegisterServiceMetadata  
legacyEndpointContextSetter  
SetLogger  
OperationInputValidation  
spanInitializeEnd  
Serialize stack step  
spanBuildRequestStart  
setOperationInput  
ResolveEndpoint  
OperationSerializer  
Build stack step  
ClientRequestID  
ComputeContentLength  
SetCredentialSourceMiddleware  
UserAgent  
AddTimeOffsetMiddleware  
RecursionDetection  
spanBuildRequestEnd  
Finalize stack step  
spanRetryLoop  
Retry  
RetryMetricsHeader  
ResolveAuthScheme  
GetIdentity  
ResolveEndpointV2  
disableHTTPS  
ComputePayloadHash  
setLegacyContextSigningOptions  
Signing  
Deserialize stack step  
AddRawResponseToMetadata  
ErrorCloseResponseBody

CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
```

```

 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
 "type": "structure",
 "members": {
 "DirectoryId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
 "traits": {
 "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
 }
 },
 "ActiveDirectorySettings": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
 "traits": {
 "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about \n\t\t\tthe Active Directory, including network access details such as domain name and IP \n\t\t\taddresses, and the credential provider for user administration.</p>"
 }
 },
 "ActiveDirectoryType": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
 "traits": {
 "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an \n\t\t\tAmazon Web Services Managed Active Directory.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
 "type": "structure",
 "members": {
 "DomainName": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
 }
 },
 "DomainIpv4List": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
 "traits": {
 "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
 "smithy.api#length": {

```

```

 "min": 1,
 "max": 2
 }
}
},
"DomainCredentialsProvider": {
 "target":
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
 "traits": {
 "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that
contains \n\t\t\tinformation about the credential provider for user administration.</p>"
 }
},
"DomainNetworkSettings": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
 "traits": {
 "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of \n\t\t\tsubnets that apply for the Active Directory.</p>"
 }
},
"traits": {
 "smithy.api#documentation": "<p>Contains network access and credential details that are needed for
\n\t\t\tuser administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType":
{
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "SELF_MANAGED",
 "value": "SELF_MANAGED",
 "documentation": "SELF_MANAGED type of Active Directory"
 },
 {
 "name": "AWS_MANAGED",
 "value": "AWS_MANAGED",
 "documentation": "AWS_MANAGED type of Active Directory"
 }
]
 }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-

```



```

 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest": {
 "type": "structure",
 "members": {
 "Username": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
 "smithy.api#required": {}
 }
 },
 "InstanceId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
 "smithy.api#required": {}
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>The identity provider for the user.</p>",
 "smithy.api#required": {}
 }
 },
 "Domain": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
 }
 },
 "Tags": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {

```







```

 "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse":
{
 "type": "structure",
 "members": {
 "IdentityProviderArn": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider
specified in the request.</p>"
 }
 },
 "LicenseServerEndpointArn": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
 "type": "union",
 "members": {
 "SecretsManagerCredentialsProvider":
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
 "traits": {
 "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for \n\t\t\t\t\tuser administration in the Active Directory.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Contains information about the credential provider for user
\n\t\t\t\t\tadministration.</p>"
 }
}

```

```

 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
]
 },
 "traits": {
 "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/license-server/DeleteLicenseServerEndpoint"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
 "type": "structure",
 "members": {
 "LicenseServerEndpointArn": {

```

```

 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> \n\t\t\tresource to delete.</p>",
 "smithy.api#required": {}
 }
 },
 "ServerType": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
 "traits": {
 "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
 "smithy.api#required": {}
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
 "type": "structure",
 "members": {
 "LicenseServerEndpoint": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
 "traits": {
 "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/identity-provider/DeregisterIdentityProvider"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest":
{
 "type": "structure",
 "members": {
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
 }
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n<p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | \n<code>REMOTE_DESKTOP_SERVICES</code>\n </p>"
 }
 }
 },

```

```

 "IdentityProviderArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
 "type": "structure",
 "members": {
 "IdentityProviderSummary": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#Directory": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^([a-z0-9-]{10})$"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {

```

```

 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/user/DisassociateUser"
 },
 "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
 "type": "structure",
 "members": {
 "Username": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the Active Directory identity provider for the
user.</p>"
 }
 },
 "InstanceId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
 }
 }
 },
 "IdentityProvider": {

```

```

 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the Active
Directory identity provider.</p>"
 }
 },
 "InstanceUserArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
 }
 },
 "Domain": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
 "type": "structure",
 "members": {
 "InstanceUserSummary":
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
 "type": "structure",
 "members": {
 "Subnets": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
 "traits": {
 "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory

```

```
domain.</p>",

 "smithy.api#length": {
 "min": 1
 },

"smithy.api#required": {}
 }
},

"traits": {
 "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
,

"com.amazonaws.licensemanagerusersubscriptions#Filter": {
 "type": "structure",
 "members": {
 "Attribute": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
 }
 },
 "Operation": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
 }
 },
 "Value": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>Value of the filter.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe\n\t\t\tor list operation. You can use filters can be used to match a set of resources by specific\n\t\t\tcriteria, such as tags, attributes, or IDs.</p>"
 }
},

"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
 }
},


```





```

 }
 },
 "IdentityProviderArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
 }
 },
 "FailureMessage": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes an identity provider.</p>"
 },
 "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
 "type": "list",
 "member": {

 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary": {
 "type": "structure",
 "members": {
 "InstanceId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
 "smithy.api#required": {}
 }
 },
 "Status": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "Products": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
 "traits": {
 "smithy.api#documentation": "<p>A list of provided user-based subscription products.</p>",

```

```

 "smithy.api#required": {}
 },
 "LastStatusCheckDate": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The date of the last status check.</p>"
 }
 },
 "StatusMessage": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
 },
 "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList":
 {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
 "type": "structure",
 "members": {
 "Username": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
 "smithy.api#required": {}
 }
 },
 "InstanceId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
 "smithy.api#required": {}
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```



```

subscriptions.</p>"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
 "smithy.api#error": "server",
 "smithy.api#httpError": 500
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
 "type": "service",
 "version": "2018-05-10",
 "operations": [
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
 },
 {

```

```

 "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
 }
],
"traits": {
 "aws.api#service": {
 "sdkId": "License Manager User Subscriptions",
 "arnNamespace": "license-manager-user-subscriptions",
 "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
 }
}

```

```

},
"aws.auth#sigv4": {
 "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
 "additionalAllowedHeaders": [
 "*",
 "content-type",
 "x-amz-content-sha256",
 "x-amz-user-agent",
 "x-amzn-platform-id",
 "x-amzn-trace-id",
 "content-length",

 "x-api-key",
 "authorization",
 "x-amz-date",
 "x-amz-security-token",
 "Access-Control-Allow-Headers",
 "Access-Control-Allow-Methods",
 "Access-Control-Allow-Origin"
],
 "additionalExposedHeaders": [
 "x-amzn-errortype",
 "x-amzn-requestid",
 "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with\n\\n\\t\\ta per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
 "version": "1.0",
 "parameters": {
 "Region": {
 "builtIn": "AWS::Region",
 "required": false,
 "documentation": "The AWS region used to dispatch the request.",
 "type": "String"
 },
 "UseDualStack": {
 "builtIn": "AWS::UseDualStack",
 "required": true,
 "default": false,
 "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
 }
 }
}

```

```

 "type": "Boolean"
 },
 "UseFIPS": {
 "builtIn": "AWS::UseFIPS",
 "required": true,
 "default": false,
 "documentation": "When true, send this request to the FIPS-compliant
regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will
return an error.",
 "type": "Boolean"
 },
 "Endpoint": {
 "builtIn": "SDK::Endpoint",
 "required": false,
 "documentation": "Override the endpoint used to send this request",
 "type": "String"
 }
},
"rules": [
 {
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Endpoint"
 }
]
 }
]
 },
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
]
 },
 {
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
 "type": "error"
 },
 {
 "conditions": [

```



```

 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 },
 "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
 "type": "error"
 },
 {
 "conditions": [],
 "endpoint": {
 "url": {
 "ref": "Endpoint"
 },
 },
 "properties": {},
 "headers": {},
 },
 "type": "endpoint"
},
"tree"
},
{
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Region"
 }
]
 }
]
},
"rules": [
 {
 "conditions": [
 {
 "fn": "aws.partition",
 },
 "argv": [
 {

```

```

 "ref": "Region"
 }
],
 "assign": "PartitionResult"
}
],
"rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 },
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsFIPS"
]
 }
]
 }
]
 }
],
 }
],

```

```

 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsDualStack"
]
 }
]
 }
],
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
],
 "type": "tree"
},
{
 "conditions": [],
 "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
 "type": "error"
}
],
"type": "tree"
},
{
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [

```

```

{
 "ref": "UseFIPS"
 },
 true
]
}
],
"rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsFIPS"
]
 },
 true
]
 }
],
 "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
 "properties": {},
 "headers": {}
 },
 {
 "type": "endpoint"
 }
],
"type": "tree"
},
{
 "conditions": [],
 "error": "FIPS is enabled but this partition does not support FIPS",
 "type": "error"
}
],

```

```

 "type": "tree"
 },
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
],
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsDualStack"
]
 }
]
 }
],
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
]
 }
]
 }
]

```

```

],
 "type": "tree"
 },
 {
 "conditions": [],
 "error": "DualStack is enabled but this partition does not support DualStack",
 "type": "error"
 }
],
"type": "tree"
},
{
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
 "conditions": [],
 "error": "Invalid Configuration: Missing Region",
 "type": "error"
}
]
},
"smithy.rules#endpointTests": {
 "testCases": [
 {
 "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
 "expect":
{
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,

```

```

 "UseDualStack": true
 },
 {
 "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {

```

```

 "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS":
true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint":
{
 "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,

```



```

 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint":
{
 "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"

```

```

 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS
and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
}
},
{
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "error": "DualStack is enabled but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
 "expect": {

```

```

 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack
disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
 }
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "error": "DualStack is enabled but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},

```

```

{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
 }
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
 "expect": {

 "endpoint": {
 "url": "https://example.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
},
{
 "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
 "expect": {
 "endpoint": {
 "url": "https://example.com"
 }
 },
 "params": {
 "UseFIPS": false,
 "UseDualStack":
false,

 "Endpoint": "https://example.com"
 }
},
{
 "documentation": "For custom endpoint with fips enabled and dualstack disabled",
 "expect": {
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
 },

```







```

"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "PROVISIONING",
 "value": "PROVISIONING",
 "documentation": ""
 },
 {
 "name": "PROVISIONING_FAILED",
 "value": "PROVISIONING_FAILED",
 "documentation": ""
 },
 {
 "name": "PROVISIONED",
 "value": "PROVISIONED",
 "documentation": ""
 },
 {
 "name": "DELETING",
 "value": "DELETING",
 "documentation": ""
 },
 {
 "name": "DELETION_FAILED",
 "value": "DELETION_FAILED",
 "documentation": ""
 }
],
 {
 "name": "DELETED",
 "value": "DELETED",
 "documentation": ""
 }
]
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "HEALTHY",
 "value": "HEALTHY",
 "documentation": ""
 },
],
 },

```



```

 {
 "name": "UNHEALTHY",
 "value": "UNHEALTHY",
 "documentation": ""
 },
 {
 "name": "NOT_APPLICABLE",
 "value": "NOT_APPLICABLE",
 "documentation": ""
 }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
 "type": "structure",
 "members": {
 "ServerType": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
 "traits": {
 "smithy.api#documentation": "<p>The type of license server.</p>",
 "smithy.api#required": {}
 }
 },
 "ServerSettings": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
 "traits": {
 "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for
your \n\t\t\tserver.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
 },

```

```

"output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
},
"errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/identity-provider/ListIdentityProviders"
 },
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "IdentityProviderSummaries"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
 "type": "structure",
 "members": {
 "MaxResults": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```



```

 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/instance/ListInstances"
 },
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "InstanceSummaries"
 }
 }
 }
}

```



token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.

```
 }
 },
 "traits": {
 "smithy.api#output": {}
 },
 "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/license-server/ListLicenseServerEndpoints"
 },
 "smithy.api#paginated": {
 "inputToken": "NextToken",
```

```
"outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "LicenseServerEndpoints"
 }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
 "type": "structure",
 "members": {
 "MaxResults": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>",
 "smithy.api#range": {
 "min": 1,
 "max": 100
 }
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p>\n\n \n <p>IdentityProviderArn</p>\n \n"
 }
 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken\n\tfrom a previously truncated response.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
 "type": "structure",
 "members": {
 "LicenseServerEndpoints": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
 "traits": {
 "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that\n\t\t\t\tcontain detailed information about the RDS License Servers that meet \n\t\t\t\tthe request criteria.</p>"
 }
 }
 }
}
```

```

 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, \n\tthere are additional elements that the service hasn't included in this request. Use this token \n\t\twith the
next request to retrieve additional objects.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
 "type":
"operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity

```



```

provider.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/user/ListProductSubscriptions"
 },
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "ProductUserSummaries"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest":
{
 "type": "structure",
 "members": {
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n
<p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> |
<code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> |
\n<code>REMOTE_DESKTOP_SERVICES</code>\n </p>"
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p>\n
\n \n <p>Status</p>\n \n \n <p>Username</p>\n
\n \n <p>Domain</p>\n \n "
 }
 }
 }
}

```

```

 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>A token to specify where to start paginating.
This is the nextToken \n\tfrom a previously truncated response.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
 "type": "structure",
 "members": {
 "ProductUserSummaries": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
 }
 }
 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, \n\tthere are additional elements that the service
hasn't included in this request. Use this token \n\t\twith the next request to retrieve additional objects.</p>"
 }
 }
},
"traits": {
 "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 }
],

```

```

 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
 "smithy.api#http": {
 "method": "GET",
 "uri": "/tags/{ResourceArn}"
 },
 "smithy.api#readonly": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
 "smithy.api#httpLabel": {},
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
 "type": "structure",
 "members": {
 "Tags": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
}

```

```

 },
 "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/user/ListUserAssociations"
 },
 "smithy.api#paginated": {
 "inputToken": "NextToken",
 "outputToken": "NextToken",
 "pageSize": "MaxResults",
 "items": "InstanceUserSummaries"
 }
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {

```

```

 "type": "structure",
 "members": {
 "InstanceId": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the EC2 instance, which
provides user-based subscriptions.</p>",
 "smithy.api#required": {}
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p>\n \n
\n <p>Status</p>\n \n \n <p>Username</p>\n \n
\n <p>Domain</p>\n \n "
 }
 },
 "NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "\n\t<p>A token to specify where to start paginating. This is the nextToken
\n\tfrom a previously truncated response.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
 "type": "structure",
 "members": {

```

```

"InstanceUserSummaries": {

"target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
 }
},
"NextToken": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, \n\tthere are additional elements that the service hasn't included in this request. Use this token \n\t\twith the next request to retrieve additional objects.</p>"
 }
},
"traits": {
 "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
 "type": "structure",
 "members": {
 "Username":
{
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
 "smithy.api#required": {}
 }
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
 "smithy.api#required": {}
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
 "smithy.api#required": {}
 }
 },
 "Status": {
 "target": "smithy.api#String",

```

```

 "traits": {
 "smithy.api#documentation": "<p>The status of a product for this user.</p>",
 "smithy.api#required": {}
 }
 },
 "ProductUserArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
 }
 },
 "StatusMessage": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The status message for a product for this user.</p>"
 }
 },
 "Domain": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
 }
 },
 "SubscriptionStartDate": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The start date of a subscription.</p>"
 }
 },
 "SubscriptionEndDate": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The end date of a subscription.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A summary of the user-based subscription products for a specific
user.</p>"
 },
 "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
 "type":
"list",
 "member": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
 }
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
 "type": "structure",
 "members": {
 "RdsSalCredentialsProvider": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
 "traits": {
 "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to \n\t\t\tthe credentials provider that's used for RDS license server user administration.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target":
"com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
 }
]
}

```



```

 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/identity-provider/RegisterIdentityProvider"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest": {
 "type": "structure",
 "members": {
 "IdentityProvider": {

 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
 "smithy.api#required": {}
 }
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n
<p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> |
<code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> |
\n<code>REMOTE_DESKTOP_SERVICES</code>\n </p>",
 "smithy.api#required": {}
 }
 },
 "Settings": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
 "traits": {
 {
 "smithy.api#documentation": "<p>The registered identity providers product related configuration
\n\t\t\tsettings such as the subnets to provision VPC endpoints.</p>"
 }
 },
 "Tags": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
 }
 }
 }
 }
}

```

```

 }
 },
 "traits": {
 "smithy.api#input": {}
 },
 "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
 "type": "structure",
 "members": {
 "IdentityProviderSummary": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
 "traits": {
 "smithy.api#documentation":
 "<p>Metadata that describes the results of an identity provider operation.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 404
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
 "type": "structure",
 "members": {
 "SecretId": {

```



```

 }
 }
},
"traits": {
 "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
 "type": "string",
 "traits": {
 "smithy.api#enum": [
 {
 "name": "RDS_SAL",
 "value": "RDS_SAL",
 "documentation": ""
 }
]
 }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
 "smithy.api#error": "client"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
 "type": "structure",
 "members": {
 "Subnets": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
 "traits": {
 "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
 "smithy.api#length": {
 "min": 1
 },
 "smithy.api#required": {}
 }
 },
 "SecurityGroupId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",

```



provider.

**Note** Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as **Pending** billing status) in Amazon Web Services Billing. For more information, see [Viewing your monthly charges](https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html) in the *Amazon Web Services Billing User Guide*.

```

 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/user/StartProductSubscription"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
 "type": "structure",
 "members": {
 "Username": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
 "smithy.api#required": {}
 }
 },
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
 "smithy.api#required": {}
 }
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n <p>Valid values:
<code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | \n<code>REMOTE_DESKTOP_SERVICES</code>\n
</p>",
 "smithy.api#required": {}
 }
 },
 "Domain": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
for whom to start the product \n\t\t\t\t\tsubscription.</p>"
 }
 }
 }
 },

```

```

 "Tags": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
 "type": "structure",
 "members": {
 "ProductUserSummary": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
 },
 {
 "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
 },
 {

```

```

 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
 "smithy.api#http": {
 "code": 200,
 "method": "POST",
 "uri": "/user/StopProductSubscription"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
 "type": "structure",
 "members": {
 "Username": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
 }
 },
 "IdentityProvider": {
 "target":
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
 "traits": {
 "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
 }
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n
<p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> |
<code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> |
\n<code>REMOTE_DESKTOP_SERVICES</code>\n </p>"
 }
 }
 },
 "documentation": "<p>An object that specifies details for the identity provider.</p>"
}

```



```

 "ProductUserArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
 }
 },
 "Domain": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
for whom to stop the product \n\t\t\tsubscription.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
 "type": "structure",
 "members": {
 "ProductUserSummary": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
 "traits": {
 "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
 "smithy.api#required": {}
 }
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
 "type": "list",
 "member": {
 "target": "smithy.api#String"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
 "type": "string",
 "traits": {
 "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
 }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
 }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
 "type": "list",
 "member": {
 "target": "smithy.api#String"
 },
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 50
 },
 "smithy.api#sensitive": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation":
 "<p>Adds tags to a resource.</p>",
 "smithy.api#http": {
 "method": "PUT",
 "uri": "/tags/{ResourceArn}"
 },
 "smithy.api#idempotent": {}
 }
}

```

```

 },
 "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
 "smithy.api#httpLabel": {},
 "smithy.api#required": {}
 }
 },
 "Tags": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
 "traits": {
 "smithy.api#documentation":
 "<p>The tags to apply to the specified resource.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#Tags": {
 "type": "map",
 "key": {
 "target": "smithy.api#String"
 },
 "value": {
 "target": "smithy.api#String"
 },
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 50
 },
 "smithy.api#sensitive": {}
 }
 }
 }

```

```

 },
 "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
 {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
 "smithy.api#error": "client"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
 "smithy.api#http": {
 "method": "DELETE",
 "uri": "/tags/{ResourceArn}"
 },
 "smithy.api#idempotent": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want

```

```

to remove tags from.</p>",
 "smithy.api#httpLabel": {},
 "smithy.api#required": {}

}
},
"TagKeys": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
 "traits": {
 "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
 "smithy.api#httpQuery": "tagKeys",
 "smithy.api#required": {}
 }
}
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
 },
 "output": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
 },
 {
 "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
 }
],

```

```

 "traits": {
 "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity\n\t\t\tprovider.</p>",
 "smithy.api#http": {

"code": 200,
 "method": "POST",
 "uri": "/identity-provider/UpdateIdentityProviderSettings"
 },
 "smithy.api#idempotent": {}
 }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
 "type": "structure",
 "members": {
 "IdentityProvider": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
 },
 "Product": {
 "target": "smithy.api#String",
 "traits": {
 "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>\n
<p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> |
<code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> |
\n<code>REMOTE_DESKTOP_SERVICES</code>\n </p>"
 }
 },
 "IdentityProviderArn": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
 }
 },
 "UpdateSettings": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
 "traits": {
 "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can\n\t\t\t\t\tupdate any combination of settings in a single operation such as the:</p>\n
\n \n <p>Subnets which you want to add to provision VPC endpoints.</p>\n \n
 \n <p>Subnets which you want to remove the VPC endpoints from.</p>\n \n
 \n <p>Security
group ID which permits traffic to the VPC endpoints.</p>\n \n ",
 "smithy.api#required": {}
 }
 }
 },

```

```

 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
 "type": "structure",
 "members": {
 "IdentityProviderSummary": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
 "traits": {
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
 "type": "structure",
 "members": {
 "AddSubnets": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
 "traits": {
 "smithy.api#documentation": "<p>The ID of one or more subnets in which License Manager will
create a VPC endpoint for products that require connectivity to activation servers.</p>",
 "smithy.api#required": {}
 }
 },
 "RemoveSubnets": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
 "traits": {
 "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
 "smithy.api#required": {}
 }
 },
 "SecurityGroupId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
 "traits": {
 "smithy.api#documentation": "<p>A security group ID that
allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for
activation servers.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration"
 }
 }
}

```

```

settings such as\n\t\tthe subnets to provision VPC endpoints.</p>"
 }
 },
 "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A parameter is not valid.</p>",
 "smithy.api#error": "client"
 }
 }
}
}
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.



"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ListReceivedLicenses

Initialize stack step

spanInitializeStart

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
Retry
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
 "context"
 "fmt"
 awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
 "github.com/aws/smithy-go/middleware"
 smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
 if params == nil {
 params = &CheckInLicenseInput{}
 }

 result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,

```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
 return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

 // License consumption token.
 //
 // This member is required.
 LicenseConsumptionToken *string

 // License beneficiary.
 Beneficiary *string

 noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
 // Metadata pertaining to the operation's result.
 ResultMetadata middleware.Metadata

 noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
 if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
 return err
 }
 err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
 if err != nil {
 return err
 }
 err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
 if err != nil {
 return err
 }
 if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
 return fmt.Errorf("add protocol finalizers: %v", err)
 }

 if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
 return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
 return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
 return err
}
if err = addRecursionDetection(stack); err != nil {
 return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
 return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
 return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
 return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
 return err
}
if err = addSpanInitializeStart(stack); err != nil {
 return err
}
if err = addSpanInitializeEnd(stack); err != nil {
 return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
 return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
 return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
 Region: region,
 ServiceID: ServiceID,
 OperationName: "CheckInLicense",
}
}

```



```

}
{
 "smithy": "2.0",
 "metadata": {
 "suppressions": [
 {
 "id": "HttpMethodSemantics",
 "namespace": "*"
 },
 {
 "id": "HttpStatusCodeSemantics",
 "namespace": "*"
 },
 {
 "id": "PaginatedTrait",
 "namespace": "*"
 },
 {
 "id": "HttpHeaderTrait",
 "namespace": "*"
 },
 {
 "id": "HttpUriConflict",
 "namespace": "*"
 },
 {
 "id": "Service",
 "namespace": "*"
 }
]
 },
 "shapes": {
 "com.amazonaws.licensemanager#AWSLicenseManager": {
 "type": "service",
 "version": "2018-08-01",
 "operations": [
 {
 "target": "com.amazonaws.licensemanager#AcceptGrant"
 },
 {
 "target": "com.amazonaws.licensemanager#CheckInLicense"
 },
 {
 "target": "com.amazonaws.licensemanager#CheckoutBorrowLicense"
 },
 {
 "target": "com.amazonaws.licensemanager#CheckoutLicense"
 }
]
 }
 }
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#CreateGrant"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateGrantVersion"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateLicense"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateLicenseConfiguration"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateLicenseVersion"
 },
 {
 "target": "com.amazonaws.licensemanager#CreateToken"
 },
 {
 "target": "com.amazonaws.licensemanager#DeleteGrant"
 },
 {
 "target": "com.amazonaws.licensemanager#DeleteLicense"
 },
 {
 "target": "com.amazonaws.licensemanager#DeleteLicenseConfiguration"
 },
 {
 "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator"
 },
 {
 "target": "com.amazonaws.licensemanager#DeleteToken"
 },
 {
 "target": "com.amazonaws.licensemanager#ExtendLicenseConsumption"
 },
 },
 {
 "target": "com.amazonaws.licensemanager#GetAccessToken"
 },
 {

```

```

 "target": "com.amazonaws.licensemanager#GetGrant"
 },
 {
 "target": "com.amazonaws.licensemanager#GetLicense"
 },
 {
 "target": "com.amazonaws.licensemanager#GetLicenseConfiguration"
 },
 {
 "target": "com.amazonaws.licensemanager#GetLicenseConversionTask"
 },
 {
 "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator"
 },
 {
 "target": "com.amazonaws.licensemanager#GetLicenseUsage"
 },
 {
 "target": "com.amazonaws.licensemanager#GetServiceSettings"
 },
 {
 "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration"
 },
 {
 "target": "com.amazonaws.licensemanager#ListDistributedGrants"
 },
 {
 "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenseConfigurations"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenseConversionTasks"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenses"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResource"
 },
 {
 "target": "com.amazonaws.licensemanager#ListLicenseVersions"
 },

```

```

{
 "target": "com.amazonaws.licensemanager#ListReceivedGrants"
},
{
 "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganization"
},
{
 "target": "com.amazonaws.licensemanager#ListReceivedLicenses"
},
{
 "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganization"
},
{
 "target": "com.amazonaws.licensemanager#ListResourceInventory"
},
{
 "target": "com.amazonaws.licensemanager#ListTagsForResource"
},
{
 "target": "com.amazonaws.licensemanager#ListTokens"
},
{
 "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration"
},
{
 "target": "com.amazonaws.licensemanager#RejectGrant"
},
{
 "target": "com.amazonaws.licensemanager#TagResource"
},
{
 "target": "com.amazonaws.licensemanager#UntagResource"
},
{
 "target": "com.amazonaws.licensemanager#UpdateLicenseConfiguration"
},
{
 "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator"
},
{
 "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource"
},
{
 "target": "com.amazonaws.licensemanager#UpdateServiceSettings"
}
},
"traits": {
 "aws.api#service": {

```

```

 "sdkId": "License Manager",
 "arnNamespace": "license-manager",
 "cloudFormationName": "LicenseManager",
 "cloudTrailEventSource": "licensemanager.amazonaws.com",
 "endpointPrefix": "license-manager"
 },
 "aws.auth#sigv4": {
 "name": "license-manager"
 },
 "aws.protocols#awsJson1_1": {},
 "smithy.api#documentation": "<p>License Manager makes it easier to manage licenses from software
vendors across multiple \n Amazon Web Services accounts and on-premises servers.</p>",
 "smithy.api#title": "AWS License Manager",
 "smithy.api#xmlNamespace": {
 "uri": "https://license-manager.amazonaws.com/doc/2018_08_01"
 },
 "smithy.rules#endpointRuleSet": {
 "version": "1.0",
 "parameters": {
 "Region": {
 "builtIn": "AWS::Region",
 "required": false,
 "documentation": "The AWS region used to dispatch the request.",
 "type": "String"
 },
 "UseDualStack": {
 "builtIn": "AWS::UseDualStack",
 "required": true,
 "default": false,
 "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
 "type": "Boolean"
 },
 "UseFIPS": {
 "builtIn": "AWS::UseFIPS",
 "required": true,
 "default": false,
 "documentation": "When true, send
this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant
endpoint, dispatching the request will return an error.",
 "type": "Boolean"
 },
 "Endpoint": {
 "builtIn": "SDK::Endpoint",
 "required": false,
 "documentation": "Override the endpoint used to send this request",
 "type": "String"
 }
 }
 }
}

```

```

 },
 "rules": [
 {
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Endpoint"
 }
]
 }
]
 }
],
 "type": "tree",
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
]
 },
 {
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
 "type": "error"
 }
],
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
],
 {
 "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
 "type": "error"
 }
],
 {

```

```

 "conditions": [],
 "endpoint": {
 "url": {
 "ref": "Endpoint"
 },
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
]
},
{
 "conditions": [
 {
 "fn": "isSet",
 "argv": [
 {
 "ref": "Region"
 }
]
 }
]
},
 "type": "tree",
 "rules": [
 {
 "conditions": [
 {
 "fn": "aws.partition",
 "argv": [
 {
 "ref": "Region"
 }
]
 },
],
 "assign": "PartitionResult"
 }
],
 "type": "tree",
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref":

```

"UseFIPS"

```

 },

```

```

 true
]
},
{
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
}
],
"type": "tree",
"rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsFIPS"
]
 }
]
 }
]
 },
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsDualStack"
]
 }
]
 }
]

```



```

 }
],
 "type": "tree",
 "rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 }
],
 "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
 "type": "error"
}
]
},
{
 "conditions": [
 {
 "fn": "booleanEquals",

 "argv": [
 {
 "ref": "UseFIPS"
 },
 true
]
 }
],
 "type": "tree",
 "rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,

```

```

{
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
 "supportsFIPS"
]
}
],
"type": "tree",
"rules": [

{
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager-fips.{Region}.{PartitionResult#dnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
}
],
{
 "conditions": [],
 "error": "FIPS is enabled but this
partition does not support FIPS",
 "type": "error"
}
],
{
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 {
 "ref": "UseDualStack"
 },
 true
]
 }
]
},
"type": "tree",

```

```

"rules": [
 {
 "conditions": [
 {
 "fn": "booleanEquals",
 "argv": [
 true,
 {
 "fn": "getAttr",
 "argv": [
 {
 "ref": "PartitionResult"
 },
],
 },
],
 },
],
 "supportsDualStack"
 },
],
"type": "tree",
"rules": [
 {
 "conditions": [],
 "endpoint": {
 "url": "https://license-
manager.{Region}.{PartitionResult#dualStackDnsSuffix}",
 "properties": {},
 "headers": {}
 },
 "type": "endpoint"
 },
],
{
 "conditions": [],
 "error": "DualStack is enabled but this partition does not support DualStack",
 "type": "error"
}
],
{
 "conditions": [],
 "endpoint": {
 "url": "https://license-manager.{Region}.{PartitionResult#dnsSuffix}",
 "properties": {},
 "headers": {}
 },

```

```

 },
 "type": "endpoint"
 }
]
}
],
},
{
 "conditions": [],
 "error": "Invalid Configuration: Missing Region",
 "type": "error"
}
]
},
"smithy.rules#endpointTests": {
 "testCases": [
 {
 "documentation": "For region af-south-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.af-south-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "af-south-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region ap-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "ap-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region ap-northeast-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-northeast-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "ap-northeast-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 }
]
}

```

```

 }
 },
 "params": {
 "Region": "ap-northeast-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ap-northeast-2 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-northeast-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "ap-northeast-2",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ap-northeast-3 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-northeast-3.amazonaws.com"
 }
 },
 "params": {
 "Region": "ap-northeast-3",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ap-south-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-south-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "ap-south-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},

```

```

{
 "documentation": "For region ap-southeast-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-southeast-1.amazonaws.com"
 }
 }
},
{
 "params": {
 "Region": "ap-southeast-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ap-southeast-2 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-southeast-2.amazonaws.com"
 }
 }
},
{
 "params": {
 "Region": "ap-southeast-2",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ap-southeast-3 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ap-southeast-3.amazonaws.com"
 }
 }
},
{
 "params": {
 "Region": "ap-southeast-3",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region ca-central-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.ca-central-1.amazonaws.com"
 }
 }
},

```

```

 "params": {
 "Region": "ca-central-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region eu-central-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-central-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "eu-central-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region eu-north-1 with FIPS
disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-north-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "eu-north-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region eu-south-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-south-1.amazonaws.com"
 }
 },
 "params": {
 "Region":
"eu-south-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },

```

```

{
 "documentation": "For region eu-west-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-west-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "eu-west-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region eu-west-2 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-west-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "eu-west-2",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region eu-west-3 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.eu-west-3.amazonaws.com"
 }
 },
 "params": {
 "Region": "eu-west-3",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region me-south-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.me-south-1.amazonaws.com"
 }
 },

```



```

 "params": {
 "Region": "me-south-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region sa-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.sa-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "sa-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-east-2 with FIPS disabled and DualStack disabled",

```

```

 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-east-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-2",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-east-2 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-east-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-east-2",
 "UseFIPS": true,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region
us-west-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-west-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-west-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-west-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-west-1.amazonaws.com"
 }
 },
 "params": {

```

```

 "Region": "us-west-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-west-2 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-west-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-west-2",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-west-2 with FIPS enabled and DualStack disabled",

"expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-west-2.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-west-2",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,

 "UseDualStack": true
 }
},
{
 "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",

```

```

 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url":
"https://license-manager.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region cn-northwest-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.cn-northwest-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-northwest-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 }
},
{
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.cn-north-1.api.amazonwebservices.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",

```

```

 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.cn-north-1.amazonaws.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.cn-north-1.api.amazonwebservices.com.cn"
 }
 },
 "params": {
 "Region": "cn-north-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-gov-east-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {

```

```

 "url": "https://license-manager-fips.us-gov-east-1.amazonaws.com"
 }
},
"params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": false
}
},
{
 "documentation": "For region us-gov-west-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-gov-west-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-west-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation":
 "For region us-gov-west-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-gov-west-1.amazonaws.com"
 }
 },
 "params": {
 "Region": "us-gov-west-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-gov-east-1.api.aws"
 }
 },
},
"params": {
 "Region": "us-gov-east-1",
 "UseFIPS": true,
 "UseDualStack": true
}

```

```

 }
 },
 {
 "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-gov-east-1.api.aws"
 }
 },
 "params": {
 "Region": "us-gov-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack
disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
 },
 {
 "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-iso-east-1.c2s.ic.gov"
 }
 },
 "params": {

```

```

 "Region": "us-iso-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "error": "DualStack is enabled
but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-iso-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager.us-isob-east-1.sc2s.sgov.gov"
 }
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack enabled",
 "expect": {
 "error": "FIPS and DualStack are enabled, but this partition does not support one or both"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": true
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
 "expect": {
 "endpoint": {
 "url": "https://license-manager-fips.us-isob-east-1.sc2s.sgov.gov"
 }
 }
}

```



```

 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": true,
 "UseDualStack": false
 }
},
{
 "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack enabled",
 "expect": {
 "error": "DualStack is enabled but this partition does not support DualStack"
 },
 "params": {
 "Region": "us-isob-east-1",
 "UseFIPS": false,
 "UseDualStack": true
 }
},
{
 "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

"expect": {
 "endpoint": {
 "url": "https://example.com"
 }
},
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
},
{
 "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
 "expect": {
 "endpoint": {
 "url": "https://example.com"
 }
 },
 "params": {
 "UseFIPS": false,

"UseDualStack": false,
 "Endpoint": "https://example.com"
 }
}

```

```

 },
 {
 "documentation": "For custom endpoint with fips enabled and dualstack disabled",
 "expect": {
 "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": true,
 "UseDualStack": false,
 "Endpoint": "https://example.com"
 }
 },
 {
 "documentation": "For custom endpoint with fips disabled and dualstack enabled",
 "expect": {
 "error":
"Invalid Configuration: Dualstack and custom endpoint are not supported"
 },
 "params": {
 "Region": "us-east-1",
 "UseFIPS": false,
 "UseDualStack": true,
 "Endpoint": "https://example.com"
 }
 },
 {
 "documentation": "Missing region",
 "expect": {
 "error": "Invalid Configuration: Missing Region"
 }
 }
],
 "version": "1.0"
}
},
"com.amazonaws.licensemanager#AcceptGrant": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#AcceptGrantRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#AcceptGrantResponse"
 },
 "errors": [
 {

```

```

 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
}

],
"traits": {
 "smithy.api#documentation": "<p>Accepts the specified grant.</p>"
}
},
"com.amazonaws.licensemanager#AcceptGrantRequest": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#AcceptGrantResponse": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Grant

```

```

ARN.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant version.</p>"
 }
 },
 },
 "traits": {
 "smithy.api#output": {}
 },
 },
 "com.amazonaws.licensemanager#AccessDeniedException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "ServiceAccessDenied",
 "httpResponseCode": 401
 },
 "smithy.api#documentation": "<p>Access to resource denied.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 401
 }
 },
 "com.amazonaws.licensemanager#ActivationOverrideBehavior": {
 "type": "enum",
 "members": {
 "DISTRIBUTED_GRANTS_ONLY": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DISTRIBUTED_GRANTS_ONLY"
 }
 },
 "ALL_GRANTS_PERMITTED_BY_ISSUER": {

```

```

 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "ALL_GRANTS_PERMITTED_BY_ISSUER"
 }
 },
 "com.amazonaws.licensemanager#AllowedOperation": {
 "type": "enum",
 "members": {
 "CREATE_GRANT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CreateGrant"
 }
 },
 "CHECKOUT_LICENSE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CheckoutLicense"
 }
 },
 "CHECKOUT_BORROW_LICENSE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CheckoutBorrowLicense"
 }
 },
 "CHECK_IN_LICENSE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CheckInLicense"
 }
 },
 "EXTEND_CONSUMPTION_LICENSE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "ExtendConsumptionLicense"
 }
 },
 "LIST_PURCHASED_LICENSES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "ListPurchasedLicenses"
 }
 }
 }
 },

```

```

 "CREATE_TOKEN": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CreateToken"
 }
 }
 },
 "com.amazonaws.licensemanager#AllowedOperationList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#AllowedOperation"
 },
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max":
7
 }
 }
 },
 "com.amazonaws.licensemanager#Arn": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 2048
 },
 "smithy.api#pattern": "^arn:aws(-(cn|us-gov|iso-b|iso-c|iso-d))?:[A-Za-z0-9][A-Za-z0-9_/.-]{0,62}::[A-Za-z0-9_/.-]{0,63}::[A-Za-z0-9_/.-]{0,63}::[A-Za-z0-9][A-Za-z0-9:/+=,@.-]{0,1023}$"
 }
 },
 "com.amazonaws.licensemanager#ArnList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Arn"
 }
 },
 "com.amazonaws.licensemanager#AuthorizationException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError":

```

```

{
 "code": "AuthorizationFailure",
 "httpResponseCode": 403
},
 "smithy.api#documentation": "<p>The Amazon Web Services user account does not have permission to
perform the action. Check the IAM\n policy associated with this account.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 403
}
},
"com.amazonaws.licensemanager#AutomatedDiscoveryInformation": {
 "type": "structure",
 "members": {
 "LastRunTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Time that automated discovery last ran.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes automated discovery.</p>"
 }
},

"com.amazonaws.licensemanager#Boolean": {
 "type": "boolean",
 "traits": {
 "smithy.api#default": false
 }
},
"com.amazonaws.licensemanager#BorrowConfiguration": {
 "type": "structure",
 "members": {
 "AllowEarlyCheckIn": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether early check-ins are allowed.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "MaxTimeToLiveInMinutes": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum time for the borrow configuration, in minutes.</p>",
 "smithy.api#required": {}
 }
 }
}

```

```

},
 "traits": {
 "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
 }
},
"com.amazonaws.licensemanager#BoxBoolean": {
 "type": "boolean"
},
"com.amazonaws.licensemanager#BoxInteger": {
 "type": "integer"
},
"com.amazonaws.licensemanager#BoxLong": {
 "type": "long"
},
"com.amazonaws.licensemanager#CheckInLicense": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CheckInLicenseRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CheckInLicenseResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 }
],
 {
 "target": "com.amazonaws.licensemanager#ConflictException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
}

```



```

 }
],
 "traits": {
 "smithy.api#documentation": "<p>Checks in the specified license. Check in a license when it is no longer
in use.</p>"
 }
},
"com.amazonaws.licensemanager#CheckInLicenseRequest":
{
 "type": "structure",
 "members": {
 "LicenseConsumptionToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License consumption token.</p>",
 "smithy.api#required": {}
 }
 },
 "Beneficiary": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License beneficiary.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#CheckInLicenseResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CheckoutBorrowLicense":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#EntitlementNotAllowedException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
 },
 {
 "target":
"com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Checks out the specified license for offline use.</p>"
 },
 "com.amazonaws.licensemanager#CheckoutBorrowLicenseRequest": {
 "type": "structure",
 "members":
 {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The license must
use the borrow consumption configuration.</p>",
 "smithy.api#required": {}
 }
 }
 }
}

```

```

 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementDataList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements. Partial checkouts are not supported.</p>",
 "smithy.api#required": {}
 }
 },
 },
 "DigitalSignatureMethod": {
 "target": "com.amazonaws.licensemanager#DigitalSignatureMethod",
 "traits": {
 "smithy.api#documentation": "<p>Digital signature method. The possible
value is JSON Web Signature (JWS) algorithm PS384.\n For more information, see RFC 7518 Digital Signature with RSASSA-PSS.</p>",
 "smithy.api#required": {}
 }
 },
 },
 "NodeId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Node ID.</p>"
 }
 },
 },
 "CheckoutMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {
 "smithy.api#documentation": "<p>Information about constraints.</p>"
 }
 },
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation":
"<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#CheckoutBorrowLicenseResponse": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
 },
 "LicenseConsumptionToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License consumption token.</p>"
 }
 },
 "EntitlementsAllowed": {
 "target": "com.amazonaws.licensemanager#EntitlementDataList",
 "traits": {
 "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
 }
 },
 "NodeId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Node ID.</p>"
 }
 },
 "SignedToken": {
 "target": "com.amazonaws.licensemanager#SignedToken",
 "traits": {
 "smithy.api#documentation": "<p>Signed token.</p>"
 }
 },
 "IssuedAt": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date and time at which the license checkout
is issued.</p>"
 }
 },
 "Expiration": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date and time at which the license checkout expires.</p>"
 }
 },
 "CheckoutMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {
 "smithy.api#documentation": "<p>Information about constraints.</p>"
 }
 }
},

```

```

 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#CheckoutLicense": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CheckoutLicenseRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CheckoutLicenseResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#NoEntitlementsAllowedException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target":
"com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Checks out the specified license.</p>\n <note>\n <p>If the
account that created the license is the same that is performing the check out, you must\n specify the account

```

```

as the beneficiary.</p>\n </note>"
 }
 },
 "com.amazonaws.licensemanager#CheckoutLicenseRequest": {
 "type": "structure",
 "members": {
 "ProductSKU": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product
SKU.</p>",
 "smithy.api#required": {}
 }
 },
 "CheckoutType": {
 "target": "com.amazonaws.licensemanager#CheckoutType",
 "traits": {
 "smithy.api#documentation": "<p>Checkout type.</p>",
 "smithy.api#required": {}
 }
 },
 "KeyFingerprint": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Key fingerprint identifying the license.</p>",
 "smithy.api#required": {}
 }
 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementDataList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements.</p>",
 "smithy.api#required": {}
 }
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "Beneficiary": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License beneficiary.</p>"
 }
 }
 }
 }
}

```

```

 }
 },
 "NodeId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Node ID.</p>"
 }
 },
 "traits": {

 "smithy.api#input": {}
 },
 "com.amazonaws.licensemanager#CheckoutLicenseResponse": {
 "type": "structure",
 "members": {
 "CheckoutType": {
 "target": "com.amazonaws.licensemanager#CheckoutType",
 "traits": {
 "smithy.api#documentation": "<p>Checkout type.</p>"
 }
 },
 "LicenseConsumptionToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License consumption token.</p>"
 }
 },
 "EntitlementsAllowed": {
 "target": "com.amazonaws.licensemanager#EntitlementDataList",
 "traits": {
 "smithy.api#documentation": "<p>Allowed license entitlements.</p>"
 }
 },
 "SignedToken": {
 "target": "com.amazonaws.licensemanager#SignedToken",
 "traits": {
 "smithy.api#documentation": "<p>Signed token.</p>"
 }
 },
 "NodeId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Node ID.</p>"
 }
 },
 "IssuedAt": {

```

```

 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date and time at which the license checkout is issued.</p>"
 }
 },
 "Expiration": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date and
time at which the license checkout expires.</p>"
 }
 },
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the checkout license.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CheckoutType": {
 "type": "enum",
 "members": {
 "PROVISIONAL": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PROVISIONAL"
 }
 },
 "PERPETUAL": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PERPETUAL"
 }
 }
 }
},
"com.amazonaws.licensemanager#ClientRequestToken": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max": 36
 }
 }
}

```



```

 },
 "com.amazonaws.licensemanager#ClientToken": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 2048
 },
 "smithy.api#pattern": "^\\S+$"
 }
 },
 "com.amazonaws.licensemanager#ConflictException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "ConflictException",
 "httpResponseCode":
409
 },
 "smithy.api#documentation": "<p>There was a conflict processing the request. Try your request
again.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 409
 }
 },
 "com.amazonaws.licensemanager#ConsumedLicenseSummary": {
 "type": "structure",
 "members": {
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Resource type of the resource consuming a license.</p>"
 }
 },
 "ConsumedLicenses": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
 }
 }
 }
 },
 },

```

```

"traits": {
 "smithy.api#documentation": "<p>Details about license consumption.</p>"
 },
},
"com.amazonaws.licensemanager#ConsumedLicenseSummaryList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ConsumedLicenseSummary"
 }
},
"com.amazonaws.licensemanager#ConsumptionConfiguration": {
 "type": "structure",
 "members": {
 "RenewType": {
 "target": "com.amazonaws.licensemanager#RenewType",
 "traits": {
 "smithy.api#documentation": "<p>Renewal frequency.</p>"
 }
 },
 "ProvisionalConfiguration": {
 "target": "com.amazonaws.licensemanager#ProvisionalConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
 }
 },
 "BorrowConfiguration": {
 "target": "com.amazonaws.licensemanager#BorrowConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Details about a borrow configuration.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about a consumption configuration.</p>"
 }
},
"com.amazonaws.licensemanager#CreateGrant": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateGrantRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateGrantResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 }
],

```

```

 {
 "target":
"com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a grant for the specified license. A grant shares the use of
license\n entitlements with a specific Amazon Web Services account, an organization, or an\n
organizational unit (OU). For more
information, see <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager in the <i>License Manager User Guide</i>.</p>"
 }
},
"com.amazonaws.licensemanager#CreateGrantRequest": {
 "type": "structure",
 "members": {
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "GrantName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant name.</p>",
 "smithy.api#required": {}
 }
 }
 }
},
"LicenseArn": {

```

```

 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 },
 "Principals": {
 "target": "com.amazonaws.licensemanager#PrincipalArnList",
 "traits": {
 "smithy.api#documentation": "<p>The grant principals. You can specify one of the following as an Amazon Resource Name (ARN):</p>\n \n \n <p>An Amazon Web Services account, which includes only the account specified.</p>\n \n \n \n \n <p>An organizational unit (OU), which includes all accounts in the OU.</p>\n \n \n \n <p>An organization, which will include all accounts across your organization.</p>\n \n ",
 "smithy.api#required": {}
 }
 },
 "HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Home Region of the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "AllowedOperations": {
 "target": "com.amazonaws.licensemanager#AllowedOperationList",
 "traits": {
 "smithy.api#documentation": "<p>Allowed operations for the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},

"com.amazonaws.licensemanager#CreateGrantResponse": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Grant ARN.</p>"
 }
 }
 },

```

```

 "Status": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant version.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CreateGrantVersion":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateGrantVersionRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateGrantVersionResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
]
}

```

```

],
 "traits": {
 "smithy.api#documentation": "<p>Creates a new version of the specified grant. For more information,
see\n <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/granted-
licenses.html\">Granted licenses in License Manager in the <i>License Manager User Guide</i>.</p>"
 }
 },
 "com.amazonaws.licensemanager#CreateGrantVersionRequest": {
 "type": "structure",
 "members": {
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
 }
 },
 "smithy.api#required": {}
 }
 },
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "GrantName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant name.</p>"
 }
 },
 "AllowedOperations": {
 "target": "com.amazonaws.licensemanager#AllowedOperationList",
 "traits": {
 "smithy.api#documentation": "<p>Allowed operations for the grant.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "StatusReason": {
 "target": "com.amazonaws.licensemanager#StatusReasonMessage",
 "traits": {
 "smithy.api#documentation": "<p>Grant status reason.</p>"
 }
 }
}

```

```

 }
 },
 "SourceVersion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Current version of the grant.</p>"
 }
 },
 "Options": {
 "target": "com.amazonaws.licensemanager#Options",
 "traits": {
 "smithy.api#documentation": "<p>The options specified for the grant.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#CreateGrantVersionResponse": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Grant ARN.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>New version of the grant.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CreateLicense": {
 "type": "operation",

```

```

"input": {
 "target": "com.amazonaws.licensemanager#CreateLicenseRequest"
},
"output": {
 "target": "com.amazonaws.licensemanager#CreateLicenseResponse"
},
"errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Creates a license.</p>"
}
},
"com.amazonaws.licensemanager#CreateLicenseConfiguration": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 }
]
}

```



```

 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a license configuration.</p>\n <p>A license configuration is an abstraction of a customer license agreement that can be\n consumed and enforced by License Manager. Components include specifications for the license\n type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n Dedicated Instance, Dedicated Host, or all of these), license affinity to host (how long a \n license must be associated with a host), and the number of licenses purchased and used.</p>"
 }
},
"com.amazonaws.licensemanager#CreateLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the license configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Description of the license configuration.</p>"
 }
 }
 },
 "LicenseCountingType": {
 "target": "com.amazonaws.licensemanager#LicenseCountingType",
 "traits": {
 "smithy.api#documentation": "<p>Dimension used to track the license inventory.</p>",
 "smithy.api#required": {}
 }
 },
 "LicenseCount": {
 "target": "com.amazonaws.licensemanager#BoxLong",

```

```

 "traits": {
 "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
 }
},
"LicenseCountHardLimit": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether hard or soft license enforcement is used. Exceeding a hard limit\n blocks the launch of new instances.</p>"
 }
},
"LicenseRules": {
 "target":
"com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>License rules. The syntax is #name=value (for example, #allowedTenancy=EC2-DedicatedHost). The available rules \n vary by dimension, as follows.</p>\n
\n \n <p>\n <code>Cores</code> dimension: <code>allowedTenancy</code> |\n <code>licenseAffinityToHost</code> |\n <code>maximumCores</code> |\n <code>minimumCores</code>\n </p>\n \n \n <p>\n <code>Instances</code> dimension: <code>allowedTenancy</code> |\n <code>maximumCores</code> |\n <code>minimumCores</code> |\n <code>maximumSockets</code> | <code>minimumSockets</code> |\n <code>maximumVcpus</code> | <code>minimumVcpus</code>\n </p>\n \n \n <p>\n <code>Sockets</code> dimension: <code>allowedTenancy</code> |\n <code>licenseAffinityToHost</code> |\n <code>maximumSockets</code> |\n <code>minimumSockets</code>\n </p>\n \n \n <p>\n <code>vCPUs</code> dimension: <code>allowedTenancy</code> |\n <code>honorVcpuOptimization</code> |\n <code>maximumVcpus</code> |\n <code>minimumVcpus</code>\n </p>\n \n \n <p>The unit for <code>licenseAffinityToHost</code> is days and the range is 1 to 180. The possible \n values for <code>allowedTenancy</code> are <code>EC2-Default</code>, <code>EC2-DedicatedHost</code>, and \n <code>EC2-DedicatedInstance</code>. The possible values for <code>honorVcpuOptimization</code> are \n <code>True</code> and <code>False</code>.</p>"
 }
},
"Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>Tags to add to the license configuration.</p>"
 }
},
"DisassociateWhenNotFound": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>When true, disassociates a resource when software is\n uninstalled.</p>"
 }
}

```

```

 }
 },
 "ProductInformationList": {
 "target": "com.amazonaws.licensemanager#ProductInformationList",
 "traits": {
 "smithy.api#documentation": "<p>Product information.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#CreateLicenseConfigurationResponse":
{
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a new license conversion task.</p>"
 }
},
"com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource you are converting the license type for.</p>",
 "smithy.api#required": {}
 }
 },
 "SourceLicenseContext": {
 "target": "com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information that identifies the license type you are converting from. \n\n For the structure of the source license, see Convert a license type using the CLI in the <i>License Manager User Guide</i>.</p>",
 "smithy.api#required": {}
 }
 },
 "DestinationLicenseContext": {
 "target": "com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information that identifies the license type you are converting to. For the structure of the destination license, see Convert a license type using the CLI in the <i>License Manager User Guide</i>.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanager#CreateLicenseConversionTaskForResourceResponse": {
 "type": "structure",
 "members": {
 "LicenseConversionTaskId": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",

 "traits": {
 "smithy.api#documentation": "<p>The ID of the created license type conversion task.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#CreateLicenseManagerReportGenerator": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {

```

```

 "target": "com.amazonaws.licensemanager#ValidationException"
 },
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a report generator.</p>"
 }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorRequest": {
 "type": "structure",
 "members": {
 "ReportGeneratorName":
 {
 "target": "com.amazonaws.licensemanager#ReportGeneratorName",
 "traits": {
 "smithy.api#documentation": "<p>Name of the report generator.</p>",
 "smithy.api#required": {}
 }
 },
 "Type": {
 "target": "com.amazonaws.licensemanager#ReportTypeList",
 "traits": {
 "smithy.api#documentation": "<p>Type of reports to generate. The following report types an be
generated:</p>\n \n \n <p>License configuration report - Reports the number and
details of consumed licenses for a license configuration.</p>\n \n \n <p>Resource
report - Reports the tracked licenses and resource consumption for a license configuration.</p>\n \n
 ",
 "smithy.api#required": {}
 }
 },
 "ReportContext": {
 "target": "com.amazonaws.licensemanager#ReportContext",
 "traits": {
 "smithy.api#documentation": "<p>Defines the type of license configuration the report generator
tracks.</p>",
 "smithy.api#required": {}
 }
 },
 "ReportFrequency": {
 "target": "com.amazonaws.licensemanager#ReportFrequency",
 "traits": {
 "smithy.api#documentation": "<p>Frequency by which reports are generated. Reports can be
generated daily, monthly, or weekly.</p>",
 "smithy.api#required": {}
 }
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientRequestToken",

```

```

 "traits": {
 "smithy.api#documentation": "<p>Unique,
case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Description of the report generator.</p>"
 }
 },
 "Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>Tags to add to the report generator.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#CreateLicenseManagerReportGeneratorResponse": {
 "type": "structure",
 "members": {
 "LicenseManagerReportGeneratorArn":
{
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the new report
generator.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#CreateLicenseRequest": {
 "type": "structure",
 "members": {
 "LicenseName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License name.</p>",
 "smithy.api#required": {}
 }
 }
 }
}

```

```

 },
 "ProductName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product
name.</p>",
 "smithy.api#required": {}
 }
 },
 "ProductSKU": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product SKU.</p>",
 "smithy.api#required": {}
 }
 },
 "Issuer": {
 "target": "com.amazonaws.licensemanager#Issuer",
 "traits": {
 "smithy.api#documentation": "<p>License issuer.</p>",
 "smithy.api#required": {}
 }
 },
 "HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Home Region for the license.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "Validity": {
 "target": "com.amazonaws.licensemanager#DatetimeRange",
 "traits": {
 "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
 "smithy.api#required": {}
 }
 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements.</p>",
 "smithy.api#required": {}
 }
 },
 "Beneficiary": {
 "target": "com.amazonaws.licensemanager#String",

```



```

 "traits": {
 "smithy.api#documentation": "<p>License beneficiary.</p>",
 "smithy.api#required": {}
 }
 },
 "ConsumptionConfiguration": {
 "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional configuration for workloads\n running with continuous connectivity. Choose a borrow configuration for workloads with offline\n usage.</p>",
 "smithy.api#required": {}
 }
 },
 "LicenseMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license.</p>"
 }
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation":
 "<p>Unique, case-sensitive identifier that you provide to ensure the idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#CreateLicenseResponse": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseStatus",
 "traits": {
 "smithy.api#documentation": "<p>License status.</p>"
 }
 }
 }
},

```

```

 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License version.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 },
 "com.amazonaws.licensemanager#CreateLicenseVersion": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateLicenseVersionRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateLicenseVersionResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#ConflictException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a new version of the specified license.</p>"
 }
 }
 }

```

```

 },
 "com.amazonaws.licensemanager#CreateLicenseVersionRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target":
"com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 },
 "LicenseName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License name.</p>",
 "smithy.api#required": {}
 }
 },
 "ProductName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product name.</p>",
 "smithy.api#required": {}
 }
 },
 "Issuer": {
 "target": "com.amazonaws.licensemanager#Issuer",
 "traits": {
 "smithy.api#documentation": "<p>License issuer.</p>",
 "smithy.api#required": {}
 }
 },
 "HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Home Region of the license.</p>",
 "smithy.api#required": {}
 }
 },
 "Validity": {
 "target": "com.amazonaws.licensemanager#DatetimeRange",
 "traits": {
 "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>",
 "smithy.api#required": {}
 }
 }
 }
 }
 }

```

```

 },
 "LicenseMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license.</p>"
 }
 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements.</p>",
 "smithy.api#required": {}
 }
 },
 "ConsumptionConfiguration": {
 "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Configuration for consumption of the license. Choose a provisional
configuration for workloads\n running with continuous connectivity. Choose a borrow configuration for
workloads with offline\n usage.</p>",
 "smithy.api#required": {}
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseStatus",
 "traits": {
 "smithy.api#documentation": "<p>License status.</p>",
 "smithy.api#required": {}
 }
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "SourceVersion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Current version of the license.</p>"
 }
 }
 },
 "traits": {

```

```

"smithy.api#input": {}
 },
 "com.amazonaws.licensemanager#CreateLicenseVersionResponse": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>License ARN.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>New version of the license.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseStatus",
 "traits": {
 "smithy.api#documentation": "<p>License status.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#CreateToken": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#CreateTokenRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#CreateTokenResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 }
]
 }
}

```

```

 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Creates a long-lived token.</p>\n <p>A refresh token is a JWT token used to get an access token. With an access token,\n you can call AssumeRoleWithWebIdentity to get role credentials that you can use to \n call License Manager to manage the specified license.</p>"
 },
 "com.amazonaws.licensemanager#CreateTokenRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license. The ARN is mapped to the aud claim of the\n JWT token.</p>",
 "smithy.api#required": {}
 }
 },
 "RoleArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the IAM roles to embed in the token. \n License Manager does not check whether the roles are in use.</p>"
 }
 },
 "ExpirationInDays": {
 "target": "com.amazonaws.licensemanager#Integer",
 "traits": {
 "smithy.api#documentation": "<p>Token expiration, in days, counted from token creation. The default is 365 days.</p>"
 }
 }
 }
 }
}

```

```

 },
 "TokenProperties": {
 "target": "com.amazonaws.licensemanager#MaxSize3StringList",
 "traits": {
 "smithy.api#documentation": "<p>Data specified by the caller to be included in the JWT token. The
data is mapped\n to the amr claim of the JWT token.</p>"
 }
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientToken",
 "traits": {
 "smithy.api#documentation": "<p>Idempotency token, valid for 10 minutes.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#input": {}
 },
 "com.amazonaws.licensemanager#CreateTokenResponse": {
 "type": "structure",
 "members": {
 "TokenId": {
 "target": "com.amazonaws.licensemanager#String",

 "traits": {
 "smithy.api#documentation": "<p>Token ID.</p>"
 }
 },
 "TokenType": {
 "target": "com.amazonaws.licensemanager#TokenType",
 "traits": {
 "smithy.api#documentation": "<p>Token type.</p>"
 }
 },
 "Token": {
 "target": "com.amazonaws.licensemanager#TokenString",
 "traits": {
 "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#DateTime": {

```

```

 "type": "timestamp"
 },
 "com.amazonaws.licensemanager#DatetimeRange": {
 "type": "structure",
 "members": {
 "Begin": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Start of the time range.</p>",
 "smithy.api#required": {}
 }
 },
 "End": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>End of the time range.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes a time range, in ISO8601-UTC format.</p>"
 }
 },
 "com.amazonaws.licensemanager#DeleteGrant": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#DeleteGrantRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#DeleteGrantResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {

```



```

 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Deletes
the specified grant.</p>"
}
},
"com.amazonaws.licensemanager#DeleteGrantRequest": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "StatusReason": {
 "target": "com.amazonaws.licensemanager#StatusReasonMessage",
 "traits": {
 "smithy.api#documentation": "<p>The Status reason for the delete request.</p>"
 }
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Current version of the
grant.</p>",
 "smithy.api#required": {}
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#DeleteGrantResponse": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Grant ARN.</p>"
 }
 }
 }
}

```

```

 },
 "Status": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation":
"<p>Grant version.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#DeleteLicense": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#ConflictException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 }
]
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 },
 {

```

```

 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Deletes the specified license.</p>"
}
},
"com.amazonaws.licensemanager#DeleteLicenseConfiguration": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Deletes the specified license configuration.</p>\n <p>You cannot delete a license configuration that is in use.</p>"
 }
},
"com.amazonaws.licensemanager#DeleteLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {

```

```

 "smithy.api#documentation": "<p>ID of the license configuration.</p>",
 "smithy.api#required": {}
 }
}
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#DeleteLicenseConfigurationResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#DeleteLicenseManagerReportGenerator": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
]
}

```

```

 }
],
 "traits": {
 "smithy.api#documentation": "<p>Deletes the specified report generator.</p>\n <p>This action deletes the report generator, which stops it from generating future reports.\n The action cannot be reversed. It has no effect on the previous reports from this generator.</p>"
 }
},
"com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorRequest": {
 "type": "structure",
 "members": {
 "LicenseManagerReportGeneratorArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to be deleted.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#DeleteLicenseManagerReportGeneratorResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#DeleteLicenseRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "SourceVersion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Current version of the license.</p>",
 "smithy.api#required": {}
 }
 }
 }
}

```

```

 {}
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#DeleteLicenseResponse": {
 "type": "structure",
 "members": {
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseDeletionStatus",
 "traits": {
 "smithy.api#documentation": "<p>License status.</p>"
 }
 },
 "DeletionDate": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date when the license is deleted.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#DeleteToken": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#DeleteTokenRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#DeleteTokenResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#RedirectException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
],
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 },
],
"traits": {
 "smithy.api#documentation": "<p>Deletes the specified token. Must be called in the license home
Region.</p>"
}
},
"com.amazonaws.licensemanager#DeleteTokenRequest": {
 "type": "structure",
 "members": {
 "TokenId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token ID.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#DeleteTokenResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#DigitalSignatureMethod": {
 "type": "enum",
 "members": {
 "JWT_PS384": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "JWT_PS384"
 }
 }
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanager#Entitlement": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement name.</p>",
 "smithy.api#required": {}
 }
 },
 "Value": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement resource. Use only
if the unit is None.</p>"
 }
 },
 "MaxCount": {
 "target": "com.amazonaws.licensemanager#Long",
 "traits": {
 "smithy.api#documentation": "<p>Maximum entitlement count. Use if the unit is not None.</p>"
 }
 },
 "Overage": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether overages are allowed.</p>"
 }
 },
 "Unit": {
 "target": "com.amazonaws.licensemanager#EntitlementUnit",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement unit.</p>",
 "smithy.api#required": {}
 }
 },
 "AllowCheckIn": {
 "target":
"com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether check-ins are allowed.</p>"
 }
 },
 "traits": {

```



```

 "smithy.api#documentation": "<p>Describes a resource entitled for use with a license.</p>"
 }
},
"com.amazonaws.licensemanager#EntitlementData": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement data name.</p>",
 "smithy.api#required": {}
 }
 },
 "Value": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement
data value.</p>"
 }
 },
 "Unit": {
 "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement data unit.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Data associated with an entitlement resource.</p>"
 }
 },
 "com.amazonaws.licensemanager#EntitlementDataList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#EntitlementData"
 }
 },
 "com.amazonaws.licensemanager#EntitlementDataUnit": {
 "type": "enum",
 "members": {
 "COUNT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Count"
 }
 }
 }
 },

```

```

"NONE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "None"
 }
},
"SECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Seconds"
 }
},
"MICROSECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Microseconds"
 }
},
"MILLISECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Milliseconds"
 }
},
"BYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bytes"
 }
},
"KILOBYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobytes"
 }
},
"MEGABYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabytes"
 }
},
"GIGABYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabytes"
 }
},

```

```

"TERABYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabytes"
 }
},
"BITS":
{
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bits"
 }
},
"KILOBITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobits"
 }
},
"MEGABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabits"
 }
},
"GIGABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabits"
 }
},
"TERABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabits"
 }
}
},
"PERCENT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Percent"
 }
},
"BYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bytes/Second"
 }
}

```

```

 }
 },
 "KILOBYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobytes/Second"
 }
 },
 "MEGABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabytes/Second"
 }
 },
 "GIGABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabytes/Second"
 }
 },
 "TERABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabytes/Second"
 }
 },
 "BITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bits/Second"
 }
 },
 "KILOBITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobits/Second"
 }
 },
 "MEGABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue":
"Megabits/Second"
 }
 },
 "GIGABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {

```

```

 "smithy.api#enumValue": "Gigabits/Second"
 }
},
"TERABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabits/Second"
 }
},
"COUNT_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Count/Second"
 }
}
},
"com.amazonaws.licensemanager#EntitlementList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Entitlement"
 }
},
"com.amazonaws.licensemanager#EntitlementNotAllowedException":
{
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The entitlement is not allowed.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
},
"com.amazonaws.licensemanager#EntitlementUnit": {
 "type": "enum",
 "members": {
 "COUNT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Count"
 }
 },
 "NONE": {
 "target": "smithy.api#Unit",

```

```

 "traits": {
 "smithy.api#enumValue": "None"
 }
 },
 "SECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Seconds"
 }
 },
 "MICROSECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Microseconds"
 }
 },
 "MILLISECONDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Milliseconds"
 }
 },
 "BYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bytes"
 }
 },
 "KILOBYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobytes"
 }
 },
 "MEGABYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabytes"
 }
 },
 "GIGABYTES": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabytes"
 }
 },
 "TERABYTES": {

```

```

 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabytes"
 }
 },
 "BITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bits"
 }
 },
 "KILOBITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobits"
 }
 },
 "MEGABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabits"
 }
 },
 "GIGABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabits"
 }
 },
 "TERABITS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabits"
 }
 },
 "PERCENT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Percent"
 }
 },
 "BYTES_PER_SECOND": {
 "target":
"smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bytes/Second"
 }
 }

```

```

 },
 "KILOBYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobytes/Second"
 }
 },
 "MEGABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabytes/Second"
 }
 },
 "GIGABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabytes/Second"
 }
 },
 "TERABYTES_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabytes/Second"
 }
 },
 "BITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Bits/Second"
 }
 },
 "KILOBITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Kilobits/Second"
 }
 },
 "MEGABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Megabits/Second"
 }
 },
 "GIGABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Gigabits/Second"
 }
 }
}

```



```

 }
 },

 "TERABITS_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Terabits/Second"
 }
 },
 "COUNT_PER_SECOND": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Count/Second"
 }
 }
}
},
"com.amazonaws.licensemanager#EntitlementUsage": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement usage name.</p>",
 "smithy.api#required": {}
 }
 },
 "ConsumedValue": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Resource usage consumed.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxCount": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Maximum entitlement usage count.</p>"
 }
 },
 "Unit": {
 "target": "com.amazonaws.licensemanager#EntitlementDataUnit",
 "traits": {
 "smithy.api#documentation": "<p>Entitlement usage unit.</p>",
 "smithy.api#required": {}
 }
 }
 }
}

```

```

 },
 "traits": {
 "smithy.api#documentation": "<p>Usage associated with an entitlement resource.</p>"
 }
 },
 "com.amazonaws.licensemanager#EntitlementUsageList":
{
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#EntitlementUsage"
 }
},
 "com.amazonaws.licensemanager#ExtendLicenseConsumption": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Extends the expiration date for license consumption.</p>"
 }
 },
 "com.amazonaws.licensemanager#ExtendLicenseConsumptionRequest": {

```

```

 "type": "structure",
 "members": {
 "LicenseConsumptionToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License consumption token.</p>",
 "smithy.api#required": {}
 }
 },
 "DryRun": {
 "target": "com.amazonaws.licensemanager#Boolean",
 "traits": {
 "smithy.api#default": false,
 "smithy.api#documentation": "<p>Checks whether you have the required permissions for the action, without actually making the request. Provides an error response if you do not have the required permissions.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ExtendLicenseConsumptionResponse": {
 "type": "structure",
 "members": {
 "LicenseConsumptionToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License consumption token.</p>"
 }
 },
 "Expiration": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Date and time at which the license consumption expires.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#FailedDependencyException": {
 "type": "structure",
 "members": {
 "Message": {

```

```

 "target": "com.amazonaws.licensemanager#Message"
 },
 "ErrorCode": {
 "target": "com.amazonaws.licensemanager#String"
 }
},
"traits": {
 "aws.protocols#awsQueryError": {
 "code": "FailedDependency",
 "httpResponseCode": 424
 },
 "smithy.api#documentation":
"<p>A dependency required to run the API is missing.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 424
}
},
"com.amazonaws.licensemanager#Filter": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#FilterName",
 "traits": {
 "smithy.api#documentation": "<p>Name of the filter. Filter names are case-sensitive.</p>"
 }
 },
 "Values": {
 "target": "com.amazonaws.licensemanager#FilterValues",
 "traits": {
 "smithy.api#documentation": "<p>The value of the filter, which is case-sensitive. You can only specify one value for the filter.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a\n describe operation. Filters can be used to match a set of resources by specific criteria,\n such as tags, attributes, or IDs.</p>"
 }
},
"com.amazonaws.licensemanager#FilterLimitExceededException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {

```

```

 "aws.protocols#awsQueryError": {
 "code": "FilterLimitExceeded",
 "httpResponseCode": 400
 },
 "smithy.api#documentation": "<p>The request uses too many filters or too many filter values.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
},
"com.amazonaws.licensemanager#FilterList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Filter"
 }
},
"com.amazonaws.licensemanager#FilterName": {
 "type": "string"
},
"com.amazonaws.licensemanager#FilterValue": {
 "type": "string"
},
"com.amazonaws.licensemanager#FilterValues": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#FilterValue",
 "traits": {
 "smithy.api#xmlName": "item"
 }
 }
},
"com.amazonaws.licensemanager#Filters": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Filter",
 "traits": {
 "smithy.api#xmlName": "item"
 }
 }
},
"com.amazonaws.licensemanager#GetAccessToken": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetAccessTokenRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetAccessTokenResponse"
 },
 "errors": [

```

```

 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets a temporary access token to use with
AssumeRoleWithWebIdentity. Access
tokens\n are valid for one hour.</p>"
 }
},
"com.amazonaws.licensemanager#GetAccessTokenRequest": {
 "type": "structure",
 "members": {
 "Token": {
 "target": "com.amazonaws.licensemanager#TokenString",
 "traits": {
 "smithy.api#documentation": "<p>Refresh token, encoded as a JWT token.</p>",
 "smithy.api#required": {}
 }
 },
 "TokenProperties": {
 "target": "com.amazonaws.licensemanager#MaxSize3StringList",
 "traits": {
 "smithy.api#documentation": "<p>Token properties to validate against those present in the JWT
token.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#GetAccessTokenResponse":
{
 "type": "structure",
 "members": {

```

```

 "AccessToken": {
 "target": "com.amazonaws.licensemanager#TokenString",
 "traits": {
 "smithy.api#documentation": "<p>Temporary access token.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 },
 "com.amazonaws.licensemanager#GetGrant": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetGrantRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetGrantResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets detailed information about the specified grant.</p>"
 }
 },
 "com.amazonaws.licensemanager#GetGrantRequest": {

```

```

"type": "structure",
"members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant version.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#GetGrantResponse": {
 "type": "structure",
 "members": {
 "Grant": {
 "target": "com.amazonaws.licensemanager#Grant",
 "traits": {
 "smithy.api#documentation": "<p>Grant details.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#GetLicense": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetLicenseRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetLicenseResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {

```



```

 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
},
{
 "target": "com.amazonaws.licensemanager#ValidationException"
}
],
"traits": {
 "smithy.api#documentation": "<p>Gets detailed information about the specified license.</p>"
}
},
"com.amazonaws.licensemanager#GetLicenseConfiguration": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets detailed information about the specified license configuration.</p>"
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanager#GetLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#GetLicenseConfigurationResponse": {
 "type": "structure",
 "members": {
 "LicenseConfigurationId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Unique ID for the license configuration.</p>"
 }
 },
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
 }
 },
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the license configuration.</p>"
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Description of the license configuration.</p>"
 }
 },
 "LicenseCountingType": {
 "target": "com.amazonaws.licensemanager#LicenseCountingType",

```

```

 "traits": {
 "smithy.api#documentation": "<p>Dimension for which the licenses are counted.</p>"
 }
 },
 "LicenseRules": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>License rules.</p>"
 }
 },
 "LicenseCount": {
 "target": "com.amazonaws.licensemanager#BoxLong",

 "traits": {
 "smithy.api#documentation": "<p>Number of available licenses.</p>"
 }
 },
 "LicenseCountHardLimit": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Sets the number of available licenses as a hard limit.</p>"
 }
 },
 "ConsumedLicenses": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of licenses assigned to resources.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License configuration status.</p>"
 }
 },
},
"OwnerAccountId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Account ID of the owner of the license configuration.</p>"
 }
},
"ConsumedLicenseSummaryList": {
 "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
 "traits": {
 "smithy.api#documentation": "<p>Summaries of the licenses consumed by resources.</p>"
 }
},

```

```

"ManagedResourceSummaryList": {
 "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",
 "traits": {
 "smithy.api#documentation": "<p>Summaries of the managed resources.</p>"
 }
},
"Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>Tags for the license configuration.</p>"
 }
},
"ProductInformationList": {
 "target": "com.amazonaws.licensemanager#ProductInformationList",
 "traits": {
 "smithy.api#documentation": "<p>Product information.</p>"
 }
},
"AutomatedDiscoveryInformation": {
 "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
 "traits": {
 "smithy.api#documentation": "<p>Automated discovery information.</p>"
 }
},
"DisassociateWhenNotFound": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>When
true, disassociates a resource when software is uninstalled.</p>"
 }
},
"traits": {
 "smithy.api#output": {}
},
"com.amazonaws.licensemanager#GetLicenseConversionTask": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetLicenseConversionTaskResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 }
],

```

```

 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets information about the specified license type conversion task.</p>"
 }
},
"com.amazonaws.licensemanager#GetLicenseConversionTaskRequest": {
 "type": "structure",
 "members": {
 "LicenseConversionTaskId": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
 "traits": {
 "smithy.api#documentation": "<p>ID of the license type conversion task to retrieve information on.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#GetLicenseConversionTaskResponse": {
 "type": "structure",
 "members": {
 "LicenseConversionTaskId": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
 "traits": {
 "smithy.api#documentation": "<p>ID of the license type conversion task.</p>"
 }
 },
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the resources the license conversion task is associated with.</p>"
 }
 }
 }
}

```

```

 }
 },
 "SourceLicenseContext": {
 "target": "com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information
about the license type converted from.</p>"
 }
 },
 "DestinationLicenseContext": {
 "target": "com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license type converted to.</p>"
 }
 },
 "StatusMessage": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
 "traits": {
 "smithy.api#documentation": "<p>Status of the license type conversion task.</p>"
 }
 },
 "StartTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Time at which the license type conversion task was started .</p>"
 }
 },
 "LicenseConversionTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Amount of time to complete the license type conversion.</p>"
 }
 },
 "EndTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Time at which the license type conversion task was
completed.</p>"
 }
 },
},

```

```

 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#GetLicenseManagerReportGenerator":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets information about the specified report generator.</p>"
 }
},
 "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorRequest": {
 "type": "structure",
 "members": {
 "LicenseManagerReportGeneratorArn": {

```

```

 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>",
 "smithy.api#required":
 }
 }
},
 "traits": {
 "smithy.api#input": {}
 }
},
 "com.amazonaws.licensemanager#GetLicenseManagerReportGeneratorResponse": {
 "type": "structure",
 "members": {
 "ReportGenerator": {
 "target": "com.amazonaws.licensemanager#ReportGenerator",
 "traits": {
 "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license configurations.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#GetLicenseRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License version.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 }
}

```



```

 },
 "com.amazonaws.licensemanager#GetLicenseResponse": {
 "type": "structure",
 "members": {
 "License": {
 "target": "com.amazonaws.licensemanager#License",
 "traits": {
 "smithy.api#documentation": "<p>License details.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#GetLicenseUsage": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetLicenseUsageRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetLicenseUsageResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "target": "com.amazonaws.licensemanager#ValidationException"
 },
 "traits": {
 "smithy.api#documentation": "<p>Gets detailed information about the usage of the specified license.</p>"
 }
 }

```

```

 },
 "com.amazonaws.licensemanager#GetLicenseUsageRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#GetLicenseUsageResponse": {
 "type": "structure",
 "members": {
 "LicenseUsage": {
 "target":
"com.amazonaws.licensemanager#LicenseUsage",
 "traits": {
 "smithy.api#documentation": "<p>License usage details.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#GetServiceSettings": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#GetServiceSettingsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#GetServiceSettingsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 }
]
 }
 }

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Gets the License Manager settings for the current Region.</p>"
 }
},
"com.amazonaws.licensemanager#GetServiceSettingsRequest": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#GetServiceSettingsResponse": {
 "type": "structure",
 "members": {
 "S3BucketArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Regional S3 bucket path for storing reports, license trail event data,
discovery data,\n and so on.</p>"
 }
 }
 },
 "SnsTopicArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>SNS topic configured to receive notifications from License
Manager.</p>"
 }
 },
 "OrganizationConfiguration": {
 "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether Organizations is integrated with License
Manager for\n cross-account discovery.</p>"
 }
 },
 "EnableCrossAccountsDiscovery": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Indicates whether cross-account discovery is enabled.</p>"
 }
 }
}

```

```

 },
 "LicenseManagerResourceShareArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource share. The License
Manager management account \n provides member accounts with access to this share.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#Grant": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 },
 "GrantName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant name.</p>",
 "smithy.api#required": {}
 }
 },
 "ParentArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Parent ARN.</p>",
 "smithy.api#required": {}
 }
 },
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>License ARN.</p>",
 "smithy.api#required": {}
 }
 },
 "GranteePrincipalArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {

```

```

"smithy.api#documentation": "<p>The grantee principal ARN.</p>",
 "smithy.api#required": {}
},
"HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Home Region of the grant.</p>",
 "smithy.api#required": {}
 }
},
"GrantStatus": {
 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>",
 "smithy.api#required": {}
 }
},
"StatusReason": {
 "target": "com.amazonaws.licensemanager#StatusReasonMessage",
 "traits": {
 "smithy.api#documentation": "<p>Grant status reason.</p>"
 }
},
"Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant version.</p>",
 "smithy.api#required": {}
 }
},
"GrantedOperations": {
 "target": "com.amazonaws.licensemanager#AllowedOperationList",
 "traits": {
 "smithy.api#documentation": "<p>Granted operations.</p>",
 "smithy.api#required": {}
 }
},
"Options": {
 "target": "com.amazonaws.licensemanager#Options",
 "traits": {
 "smithy.api#documentation": "<p>The options specified for the grant.</p>"
 }
},
"traits": {
 "smithy.api#documentation":

```

```

"<p>Describes a grant.</p>"
 }
},
"com.amazonaws.licensemanager#GrantList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Grant"
 }
},
"com.amazonaws.licensemanager#GrantStatus": {
 "type": "enum",
 "members": {
 "PENDING_WORKFLOW": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_WORKFLOW"
 }
 },
 "PENDING_ACCEPT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_ACCEPT"
 }
 },
 "REJECTED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "REJECTED"
 }
 }
 },
 "ACTIVE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "ACTIVE"
 }
 },
 "FAILED_WORKFLOW": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "FAILED_WORKFLOW"
 }
 },
 "DELETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DELETED"
 }
 }
}

```

```

 },
 "PENDING_DELETE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_DELETE"
 }
 },
 "DISABLED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DISABLED"
 }
 },
 "WORKFLOW_COMPLETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "WORKFLOW_COMPLETED"
 }
 }
 }
},
"com.amazonaws.licensemanager#GrantedLicense": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
 }
 },
 "LicenseName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License name.</p>"
 }
 }
 },
 "ProductName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product name.</p>"
 }
 },
 "ProductSKU": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product SKU.</p>"
 }
 }
}

```

```

 }
 },
 "Issuer": {
 "target": "com.amazonaws.licensemanager#IssuerDetails",
 "traits": {
 "smithy.api#documentation": "<p>Granted license issuer.</p>"
 }
 },
 "HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Home Region of the granted license.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseStatus",
 "traits": {
 "smithy.api#documentation": "<p>Granted license status.</p>"
 }
 },
 "Validity": {
 "target": "com.amazonaws.licensemanager#DatetimeRange",
 "traits": {
 "smithy.api#documentation": "<p>Date and time range during which the granted license is valid, in
ISO8601-UTC format.</p>"
 }
 },
 "Beneficiary": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Granted license beneficiary.</p>"
 }
 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements.</p>"
 }
 },
 "ConsumptionConfiguration": {
 "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
 }
 },
 "LicenseMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",

```



```

 "traits": {
 "smithy.api#documentation": "<p>Granted license metadata.</p>"
 }
 },
 "CreateTime": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Creation time of the granted license.</p>"
 }
 }
},
"Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Version of the granted license.</p>"
 }
},
"ReceivedMetadata": {
 "target": "com.amazonaws.licensemanager#ReceivedMetadata",
 "traits": {
 "smithy.api#documentation": "<p>Granted license received metadata.</p>"
 }
},
"traits": {
 "smithy.api#documentation": "<p>Describes a license that is granted to a grantee.</p>"
},
"com.amazonaws.licensemanager#GrantedLicenseList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#GrantedLicense"
 }
},
"com.amazonaws.licensemanager#ISO8601DateTime":
{
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 50
 },
 "smithy.api#pattern": "^(-(?(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|0[1-9]|[1-2][0-9])T(2[0-3]|0-1[0-9]):([0-5][0-9]):([0-5][0-9])(\\.([0-9]+)?(Z|[-+](?:2[0-3]|0-1[0-9]):[0-5][0-9])+$"
 }
},
"com.amazonaws.licensemanager#Integer": {
 "type": "integer"
}

```

```

 },
 "com.amazonaws.licensemanager#InvalidParameterValueException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "InvalidParameterValueProvided",
 "httpResponseCode": 400
 },
 "smithy.api#documentation":
 "<p>One or more parameter values are not valid.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
 },
 "com.amazonaws.licensemanager#InvalidResourceStateException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "InvalidResourceState",
 "httpResponseCode": 400
 },
 "smithy.api#documentation": "<p>License Manager cannot allocate a license to a resource because of its
state. </p>\n <p>For example, you cannot allocate a license to an instance in the process of shutting\n
down.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
 },
 "com.amazonaws.licensemanager#InventoryFilter": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the filter.</p>",
 "smithy.api#required": {}
 }
 }
 }
 }
 }

```

```

 },
 "Condition": {
 "target": "com.amazonaws.licensemanager#InventoryFilterCondition",
 "traits": {
 "smithy.api#documentation": "<p>Condition of the filter.</p>",
 "smithy.api#required": {}
 }
 },
 "Value": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Value of the filter.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>An inventory filter.</p>"
 },
 "com.amazonaws.licensemanager#InventoryFilterCondition": {
 "type": "enum",
 "members": {
 "EQUALS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "EQUALS"
 }
 },
 "NOT_EQUALS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "NOT_EQUALS"
 }
 },
 "BEGINS_WITH": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "BEGINS_WITH"
 }
 },
 "CONTAINS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "CONTAINS"
 }
 }
 }
 }
}

```

```

 },
 "com.amazonaws.licensemanager#InventoryFilterList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#InventoryFilter"
 }
 },
 "com.amazonaws.licensemanager#Issuer": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Issuer name.</p>",
 "smithy.api#required": {}
 }
 },
 "SignKey": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage
of sign and verify, \n and support the RSASSA-PSS SHA-256 signing algorithm.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about the issuer of a license.</p>"
 }
 },
 "com.amazonaws.licensemanager#IssuerDetails": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Issuer name.</p>"
 }
 },
 "SignKey": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Asymmetric KMS key from Key Management Service. The KMS
key must have a key usage of sign and verify, \n and support the RSASSA-PSS SHA-256 signing
algorithm.</p>"
 }
 },
 "KeyFingerprint": {

```

```

 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Issuer key fingerprint.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details associated with the issuer of a license.</p>"
 }
},
"com.amazonaws.licensemanager#License": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
 }
 },
 "LicenseName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License name.</p>"
 }
 },
 "ProductName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product name.</p>"
 }
 },
 "ProductSKU": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Product SKU.</p>"
 }
 },
 "Issuer": {
 "target": "com.amazonaws.licensemanager#IssuerDetails",
 "traits": {
 "smithy.api#documentation": "<p>License issuer.</p>"
 }
 },
 "HomeRegion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Home Region of the license.</p>"
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseStatus",
 "traits": {
 "smithy.api#documentation": "<p>License status.</p>"
 }
 },
 "Validity": {
 "target": "com.amazonaws.licensemanager#DatetimeRange",
 "traits": {
 "smithy.api#documentation": "<p>Date and time range during which the license is valid, in ISO8601-
UTC format.</p>"
 }
 },
 "Beneficiary": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License beneficiary.</p>"
 }
 },
 "Entitlements": {
 "target": "com.amazonaws.licensemanager#EntitlementList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlements.</p>"
 }
 },
 "ConsumptionConfiguration": {
 "target": "com.amazonaws.licensemanager#ConsumptionConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Configuration for consumption of the license.</p>"
 }
 },
 "LicenseMetadata": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {
 "smithy.api#documentation": "<p>License metadata.</p>"
 }
 },
 "CreateTime": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>License
creation time.</p>"
 }
 },

```

```

"Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>License version.</p>"
 }
},
"traits": {
 "smithy.api#documentation": "<p>Software license that is managed in License Manager.</p>"
},
"com.amazonaws.licensemanager#LicenseConfiguration": {
 "type": "structure",
 "members": {
 "LicenseConfigurationId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Unique ID of the license configuration.</p>"
 }
 },
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>"
 }
 },
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the license configuration.</p>"
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Description of the license configuration.</p>"
 }
 },
 "LicenseCountingType": {
 "target": "com.amazonaws.licensemanager#LicenseCountingType",
 "traits": {
 "smithy.api#documentation": "<p>Dimension to use to track the license inventory.</p>"
 }
 },
 "LicenseRules": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>License rules.</p>"
 }
 }
 }
}

```

```

 }
 },
 "LicenseCount": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of licenses managed by the license configuration.</p>"
 }
 },
 "LicenseCountHardLimit": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Number of available licenses as a hard limit.</p>"
 }
 },
 "DisassociateWhenNotFound": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
 }
 },
 "ConsumedLicenses": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of licenses consumed. </p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Status of the license configuration.</p>"
 }
 },
 "OwnerAccountId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Account ID of the license configuration's owner.</p>"
 }
 },
 "ConsumedLicenseSummaryList": {
 "target": "com.amazonaws.licensemanager#ConsumedLicenseSummaryList",
 "traits": {
 "smithy.api#documentation": "<p>Summaries for licenses consumed by various resources.</p>"
 }
 },
 "ManagedResourceSummaryList": {
 "target": "com.amazonaws.licensemanager#ManagedResourceSummaryList",

```



```

 "traits": {
 "smithy.api#documentation": "<p>Summaries for managed resources.</p>"
 }
 },
 "ProductInformationList": {
 "target": "com.amazonaws.licensemanager#ProductInformationList",
 "traits": {
 "smithy.api#documentation": "<p>Product information.</p>"
 }
 },
 "AutomatedDiscoveryInformation": {
 "target": "com.amazonaws.licensemanager#AutomatedDiscoveryInformation",
 "traits": {
 "smithy.api#documentation": "<p>Automated discovery information.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>A license configuration is an abstraction of a customer license agreement that can be\n consumed and enforced by License Manager. Components include specifications for the license\n type (licensing by instance, socket, CPU, or vCPU), allowed tenancy (shared tenancy, \n Dedicated Instance, Dedicated Host, or all of these), host affinity (how long a VM\n must be associated with a host), and the number of licenses purchased and used.</p>"
 },
 "com.amazonaws.licensemanager#LicenseConfigurationAssociation": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
 }
 },
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Type of server resource.</p>"
 }
 },
 "ResourceOwnerId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the resource consuming licenses.</p>"
 }
 }
 }
 }
}

```

```

 },
 "AssociationTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Time when the license configuration was associated with the resource.</p>"
 }
 },
 "AmiAssociationScope": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-account</code>.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes an association with a license configuration.</p>"
 },
 "com.amazonaws.licensemanager#LicenseConfigurationAssociations": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociation"
 }
 },
 "com.amazonaws.licensemanager#LicenseConfigurationStatus": {
 "type": "enum",
 "members": {
 "AVAILABLE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "AVAILABLE"
 }
 },
 "DISABLED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DISABLED"
 }
 }
 }
 },
 "com.amazonaws.licensemanager#LicenseConfigurationUsage": {
 "type": "structure",
 "members": {
 "ResourceArn": {

```

```

 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
 }
 },
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Type of resource.</p>"
 }
 },
 "ResourceStatus": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Status of the resource.</p>"
 }
 },
 "ResourceOwnerId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
 }
 },
 "AssociationTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Time when the license configuration was initially associated with
the resource.</p>"
 }
 },
 "ConsumedLicenses": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of licenses consumed by the resource.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about the usage of a resource associated with a license
configuration.</p>"
 },
 "com.amazonaws.licensemanager#LicenseConfigurationUsageList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurationUsage"
 }
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanager#LicenseConfigurations": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseConfiguration"
 }
 },
 "com.amazonaws.licensemanager#LicenseConversionContext": {
 "type": "structure",
 "members": {
 "UsageOperation": {
 "target": "com.amazonaws.licensemanager#UsageOperation",
 "traits": {
 "smithy.api#documentation": "<p>The Usage operation value that corresponds to the license type you
are converting your resource from. For more information about which platforms correspond to which usage
operation values see <a href='\"https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-info-
fields.html#billing-info\">Sample data: usage operation by platform\\n \\n </p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
 }
 },
 "com.amazonaws.licensemanager#LicenseConversionTask": {
 "type": "structure",
 "members": {
 "LicenseConversionTaskId": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskId",
 "traits": {
 "smithy.api#documentation": "<p>The ID of the license type conversion task.</p>"
 }
 },
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource associated
with the license type\\n conversion task.</p>"
 }
 }
 },
 "SourceLicenseContext": {
 "target": "com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license type this conversion task converted
from.</p>"
 }
 }
 }
}

```

```

 }
 },
 "DestinationLicenseContext": {
 "target":
"com.amazonaws.licensemanager#LicenseConversionContext",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license type this conversion task converted
to.</p>"
 }
 },
 "Status": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTaskStatus",
 "traits": {
 "smithy.api#documentation": "<p>The status of the conversion task.</p>"
 }
 },
 "StatusMessage": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The status message for the conversion task.</p>"
 }
 },
 "StartTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation":
"<p>The time the conversion task was started at.</p>"
 }
 },
 "LicenseConversionTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>The time the usage operation value of the resource was
changed.</p>"
 }
 },
 "EndTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>The time the conversion task was completed.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Information about a license type conversion task.</p>"
 }
},
"com.amazonaws.licensemanager#LicenseConversionTaskId": {

```

```

 "type": "string",
 "traits": {

 "smithy.api#length": {
 "min": 0,
 "max": 50
 },
 "smithy.api#pattern": "^\\c-[a-zA-Z0-9]*$"
 }
},
"com.amazonaws.licensemanager#LicenseConversionTaskStatus": {
 "type": "enum",
 "members": {
 "IN_PROGRESS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "IN_PROGRESS"
 }
 },
 "SUCCEEDED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "SUCCEEDED"
 }
 },
 "FAILED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "FAILED"
 }
 }
 }
},
"com.amazonaws.licensemanager#LicenseConversionTasks":
{
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTask"
 }
},
"com.amazonaws.licensemanager#LicenseCountingType": {
 "type": "enum",
 "members": {
 "VCPU": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "vCPU"
 }
 }
 }
}

```

```

 },
 "INSTANCE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Instance"
 }
 },
 "CORE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Core"
 }
 },
 "SOCKET": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Socket"
 }
 }
 }
},
"com.amazonaws.licensemanager#LicenseDeletionStatus": {
 "type": "enum",
 "members": {
 "PENDING_DELETE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_DELETE"
 }
 }
 },
 "DELETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DELETED"
 }
 }
},
"com.amazonaws.licensemanager#LicenseList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#License"
 }
},
"com.amazonaws.licensemanager#LicenseOperationFailure": {
 "type": "structure",
 "members": {

```

```

"ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
 }
},
"ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Resource type.</p>"
 }
},
"ErrorMessage": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Error message.</p>"
 }
},
"FailureTime": {
 "target": "com.amazonaws.licensemanager#DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Failure time.</p>"
 }
},
"OperationName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the operation.</p>"
 }
},
"ResourceOwnerId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>ID of the Amazon Web Services account that owns the
resource.</p>"
 }
},
"OperationRequestedBy": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The requester is \"License Manager Automated Discovery\".</p>"
 }
},
"MetadataList": {
 "target": "com.amazonaws.licensemanager#MetadataList",
 "traits": {

```



```

 "smithy.api#documentation": "<p>Reserved.</p>"
 }
}
},
"traits": {
 "smithy.api#documentation": "<p>Describes the failure of a license operation.</p>"
}
},
"com.amazonaws.licensemanager#LicenseOperationFailureList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseOperationFailure"
 }
},
"com.amazonaws.licensemanager#LicenseSpecification": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

 "smithy.api#required": {}
 }
 },
 "AmiAssociationScope": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Scope of AMI associations. The possible value is <code>cross-
account</code>.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details for associating a license configuration with a resource.</p>"
 }
},
"com.amazonaws.licensemanager#LicenseSpecifications": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#LicenseSpecification"
 }
},
"com.amazonaws.licensemanager#LicenseStatus": {
 "type": "enum",
 "members": {
 "AVAILABLE": {

```

```

 "target": "smithy.api#Unit",

 "traits": {
 "smithy.api#enumValue": "AVAILABLE"
 }
},
"PENDING_AVAILABLE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_AVAILABLE"
 }
},
"DEACTIVATED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DEACTIVATED"
 }
},
"SUSPENDED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "SUSPENDED"
 }
},
"EXPIRED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "EXPIRED"
 }
},

"PENDING_DELETE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_DELETE"
 }
},
"DELETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DELETED"
 }
}
},
"com.amazonaws.licensemanager#LicenseUsage": {
 "type": "structure",
 "members": {

```

```

 "EntitlementUsages": {
 "target": "com.amazonaws.licensemanager#EntitlementUsageList",
 "traits": {
 "smithy.api#documentation": "<p>License entitlement usages.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes the entitlement usage associated with a license.</p>"
 }
 },
 "com.amazonaws.licensemanager#LicenseUsageException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "LicenseUsageFailure",
 "httpResponseCode": 412
 },
 "smithy.api#documentation": "<p>You do not have enough licenses available to support a new resource
launch.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 412
 }
 },
 "com.amazonaws.licensemanager#ListAssociationsForLicenseConfiguration": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
 }
]
 }
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Lists the resource associations for the
specified license configuration.</p>\n <p>Resource associations need not consume licenses from a license
configuration. \n For example, an AMI or a stopped instance might not consume a license (depending on \n
the license rules).</p>"
}
},
"com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a license configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results
to return in a single call.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListAssociationsForLicenseConfigurationResponse": {
 "type": "structure",

```

```

"members": {
 "LicenseConfigurationAssociations": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurationAssociations",
 "traits": {
 "smithy.api#documentation": "<p>Information about the associations for the license
configuration.</p>"
 }
 },
 "NextToken": {

"target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListDistributedGrants": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListDistributedGrantsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListDistributedGrantsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {

```

```

 "target": "com.amazonaws.licensemanager#ValidationException"
 },
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the grants distributed for the specified license.</p>"
 },
 },
 "com.amazonaws.licensemanager#ListDistributedGrantsRequest": {
 "type": "structure",
 "members": {
 "GrantArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {

 "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n \n \n <p>\n <code>LicenseArn</code>\n </p>\n \n \n <p>\n <code>GrantStatus</code>\n </p>\n \n \n <p>\n <code>GranteePrincipalARN</code>\n </p>\n \n \n <p>\n <code>ProductSKU</code>\n </p>\n \n \n <p>\n <code>LicenseIssuerName</code>\n </p>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListDistributedGrantsResponse": {
 "type": "structure",

```

```

"members": {
 "Grants": {
 "target": "com.amazonaws.licensemanager#GrantList",

"traits": {
 "smithy.api#documentation": "<p>Distributed grant details.</p>"
 },
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 },
 },
 "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperations": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse"
 },
 "errors": [
 {

"target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the license configuration operations that failed.</p>"
 }
 }

```

```

 },
 "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name of the license configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListFailuresForLicenseConfigurationOperationsResponse":
 {
 "type": "structure",
 "members": {
 "LicenseOperationFailureList": {
 "target": "com.amazonaws.licensemanager#LicenseOperationFailureList",
 "traits": {
 "smithy.api#documentation": "<p>License configuration operations that failed.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 }

```



```

 },
 "com.amazonaws.licensemanager#ListLicenseConfigurations": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicenseConfigurationsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the license configurations
for your account.</p>"
 }
 },
 "com.amazonaws.licensemanager#ListLicenseConfigurationsRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArns": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the license
configurations.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next
set of results.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#Filters",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n are supported:</p>\n \n \n <p>\n <code>licenseCountingType</code> - The dimension for which licenses are counted.\n Possible values are
 <code>vCPU</code> | <code>Instance</code> | <code>Core</code> | <code>Socket</code>.\n Logical
 operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n \n \n <p>\n <code>enforceLicenseCount</code> - A Boolean value that indicates whether hard license
 enforcement is used. \n Logical operators are <code>EQUALS</code> |
 <code>NOT_EQUALS</code>.</p>\n \n \n <p>\n <code>usagelimitExceeded</code> - A Boolean value that
 indicates whether the available licenses have been exceeded. \n Logical operators are
 <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n \n "
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListLicenseConfigurationsResponse": {
 "type": "structure",
 "members": {
 "LicenseConfigurations": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurations",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license configurations.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",

 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },

```

```

 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListLicenseConversionTasks": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the license type conversion tasks for your account.</p>"
 }
 },
 "com.amazonaws.licensemanager#ListLicenseConversionTasksRequest": {
 "type": "structure",
 "members": {
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 }
 }
}

```

```

 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#Filters",
 "traits": {
 "smithy.api#documentation": "<p>\n Filters to scope the results. Valid filters are
<code>ResourceArns</code> and <code>Status</code>.\n </p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListLicenseConversionTasksResponse": {
 "type": "structure",
 "members": {
 "LicenseConversionTasks": {
 "target": "com.amazonaws.licensemanager#LicenseConversionTasks",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license configuration tasks for your
account.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListLicenseManagerReportGenerators": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target":

```

```

"com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
},
"traits": {
 "smithy.api#documentation": "<p>Lists the report generators for your account.</p>"
}
},
"com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsRequest":
{
 "type": "structure",
 "members": {
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:
</p>\n \n \n <p>\n <code>LicenseConfigurationArn</code>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 }
}

```

```

 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListLicenseManagerReportGeneratorsResponse": {
 "type": "structure",
 "members": {
 "ReportGenerators": {
 "target": "com.amazonaws.licensemanager#ReportGeneratorList",
 "traits": {
 "smithy.api#documentation": "<p>A report generator that creates periodic reports about your license configurations.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {

```

```

 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Describes the license configurations for the specified resource.</p>"
}
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of a resource that has an
associated license configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListLicenseSpecificationsForResourceResponse": {
 "type": "structure",
 "members": {
 "LicenseSpecifications": {
 "target": "com.amazonaws.licensemanager#LicenseSpecifications",
 "traits": {
 "smithy.api#documentation": "<p>License configurations associated with a resource.</p>"
 }
 }
 },
 "traits": {

```

```

 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListLicenseVersions": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicenseVersionsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicenseVersionsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists all versions of the specified license.</p>"
 }
 },
 "com.amazonaws.licensemanager#ListLicenseVersionsRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>",
 "smithy.api#required": {}
 }
 }
 }
 }
}

```



```

 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#ListLicenseVersionsResponse": {
 "type": "structure",
 "members": {
 "Licenses": {
 "target": "com.amazonaws.licensemanager#LicenseList",
 "traits": {
 "smithy.api#documentation": "<p>License details.</p>"
 }
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListLicenses": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListLicensesRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListLicensesResponse"
 },

```

```

"errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Lists the licenses for your account.</p>"
},
"com.amazonaws.licensemanager#ListLicensesRequest": {
 "type": "structure",
 "members": {
 "LicenseArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
 }
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are supported:</p>\n
 \n
 \n
 <p>\n
 <code>Beneficiary</code>\n
 \n
 \n
 <p>\n
 <code>ProductSKU</code>\n
 </p>\n
 \n
 \n
 <p>\n
 <code>Fingerprint</code>\n
 </p>\n
 \n
 \n
 <p>\n
 <code>Status</code>\n
 </p>\n
 \n
 "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token

```

for the next set of results.</p>"

```
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#ListLicensesResponse": {
 "type": "structure",
 "members": {
 "Licenses": {
 "target": "com.amazonaws.licensemanager#LicenseList",
 "traits": {
 "smithy.api#documentation": "<p>License details.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListReceivedGrants": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListReceivedGrantsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListReceivedGrantsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
```

```

 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Lists grants that are received. Received grants are grants created while specifying the\n recipient as this Amazon Web Services account, your organization, or an organizational unit\n (OU) to which this member account belongs.</p>"
},
"com.amazonaws.licensemanager#ListReceivedGrantsForOrganization": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
],

```

```

 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target":
"com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the grants received for all accounts in the organization.</p>"
 }
},
"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationRequest": {
 "type": "structure",
 "members": {
 "LicenseArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the received license.</p>",
 "smithy.api#required": {}
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n
 \n
 \n
 <p>\n
 <code>ParentArn</code>\n
 </p>\n
 \n
 \n
 <p>\n
 <code>GranteePrincipalArn</code>\n
 </p>\n
 \n
 "
 }
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},

```

```

"com.amazonaws.licensemanager#ListReceivedGrantsForOrganizationResponse":
{
 "type": "structure",
 "members": {
 "Grants": {
 "target": "com.amazonaws.licensemanager#GrantList",
 "traits": {
 "smithy.api#documentation": "<p>Lists the grants the organization has received.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListReceivedGrantsRequest": {
 "type": "structure",
 "members": {
 "GrantArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the grants.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n \n \n <p>\n <code>ProductSKU</code>\n </p>\n \n \n <p>\n <code>LicenseIssuerName</code>\n </p>\n \n \n <p>\n <code>LicenseArn</code>\n </p>\n \n \n <p>\n <code>GrantStatus</code>\n </p>\n \n \n <p>\n <code>GranterAccountId</code>\n </p>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 },

```

```

 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListReceivedGrantsResponse": {
 "type": "structure",
 "members": {
 "Grants": {
 "target": "com.amazonaws.licensemanager#GrantList",
 "traits": {
 "smithy.api#documentation": "<p>Received grant details.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListReceivedLicenses": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListReceivedLicensesRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListReceivedLicensesResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 }
]
 }
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists received licenses.</p>"
 }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganization":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {

```



```

 "target": "com.amazonaws.licensemanager#ValidationException"
 },
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the licenses received for all accounts in the organization.</p>"
 }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationRequest": {
 "type": "structure",
 "members": {
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n \n \n <p>\n <code>Beneficiary</code>\n </p>\n \n \n <p>\n <code>ProductSKU</code>\n </p>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListReceivedLicensesForOrganizationResponse": {
 "type": "structure",
 "members": {
 "Licenses": {
 "target": "com.amazonaws.licensemanager#GrantedLicenseList",
 "traits": {
 "smithy.api#documentation": "<p>Lists the licenses the organization has received.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
}
},
"traits": {
 "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#ListReceivedLicensesRequest": {
 "type": "structure",
 "members": {
 "LicenseArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARNs) of the licenses.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters are
supported:</p>\n \n \n <p>\n <code>ProductSKU</code>\n </p>\n \n \n <p>\n <code>Status</code>\n </p>\n \n \n <p>\n <code>Fingerprint</code>\n </p>\n \n \n <p>\n <code>IssuerName</code>\n </p>\n \n \n <p>\n <code>Beneficiary</code>\n </p>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#ListReceivedLicensesResponse": {
 "type": "structure",

```

```

"members": {
 "Licenses": {
 "target": "com.amazonaws.licensemanager#GrantedLicenseList",
 "traits": {
 "smithy.api#documentation": "<p>Received license details.</p>"
 }
 },

 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListResourceInventory": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListResourceInventoryRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListResourceInventoryResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#FailedDependencyException"
 },
 {
 "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
]
}

```

```

 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists resources managed using Systems Manager inventory.</p>"
 }
},
"com.amazonaws.licensemanager#ListResourceInventoryRequest": {
 "type": "structure",
 "members": {
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#InventoryFilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n are supported:</p>\n \n \n <p>\n <code>account_id</code>
- The ID of the Amazon Web Services account that owns the resource.\n Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
 \n \n <p>\n <code>application_name</code> - The name of the
application. \n Logical operators are <code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
 \n \n <p>\n <code>license_included</code> - The type of license included. \n
 Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.\n Possible values
are <code>sql-server-enterprise</code> | \n <code>sql-server-standard</code> | \n <code>sql-
server-web</code> | \n <code>windows-server-datacenter</code>.</p>\n
 \n \n <p>\n <code>platform</code> - The platform of the resource.\n Logical operators are
<code>EQUALS</code> | <code>BEGINS_WITH</code>.</p>\n
 \n \n <p>\n <code>resource_id</code> - The ID of the resource. \n Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n
 \n \n <p>\n <code>tag:<key></code> - The key/value combination of a tag assigned \n to the resource. Logical
operators are <code>EQUALS</code> (single account) or\n <code>EQUALS</code> |
<code>NOT_EQUALS</code> (cross account).</p>\n
 \n "
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},

```

```

"com.amazonaws.licensemanager#ListResourceInventoryResponse": {
 "type": "structure",
 "members": {
 "ResourceInventoryList": {
 "target": "com.amazonaws.licensemanager#ResourceInventoryList",
 "traits": {
 "smithy.api#documentation": "<p>Information
about the resources.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#ListTagsForResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListTagsForResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListTagsForResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists the tags for the specified license configuration.</p>"
 }
}

```

```

 }
 },
 "com.amazonaws.licensemanager#ListTagsForResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",

 "smithy.api#required": {}
 }
 },
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListTagsForResourceResponse": {
 "type": "structure",
 "members": {
 "Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>Information about the tags.</p>"
 }
 },
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListTokens": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListTokensRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListTokensResponse"
 },
 "errors":
 [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 }
]

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists your tokens.</p>"
 }
},
"com.amazonaws.licensemanager#ListTokensRequest": {
 "type": "structure",
 "members": {
 "TokenIds": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>Token IDs.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#FilterList",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filter is supported:</p>\n
\n \n <p>\n <code>LicenseArns</code>\n </p>\n \n "
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#MaxSize100",
 "traits": {
 "smithy.api#documentation":
 "<p>Maximum number of results to return in a single call.</p>"
 }
 }
 },
 "traits": {

```

```

 "smithy.api#input": {}
 },
 "com.amazonaws.licensemanager#ListTokensResponse": {
 "type": "structure",
 "members": {
 "Tokens": {
 "target": "com.amazonaws.licensemanager#TokenList",
 "traits": {
 "smithy.api#documentation": "<p>Received token details.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#ListUsageForLicenseConfiguration":
{
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#FilterLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 }
],

```



```

 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Lists all license usage records for a license configuration, displaying
license\n consumption details by resource at a selected point in time. Use this action to audit the\n current
license consumption for any license inventory and configuration.</p>"
 }
},
"com.amazonaws.licensemanager#ListUsageForLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "MaxResults": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum number of results to return in a single call.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 },
 "Filters": {
 "target": "com.amazonaws.licensemanager#Filters",
 "traits": {
 "smithy.api#documentation": "<p>Filters to scope the results. The following filters and logical
operators\n are supported:</p>\n \n \n <p>\n <code>resourceArn</code> - The ARN of the license configuration resource.\n Logical operators are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n \n <p>\n <code>resourceType</code> - The resource type
(<code>EC2_INSTANCE</code> | <code>EC2_HOST</code> | <code>EC2_AMI</code> |
<code>SYSTEMS_MANAGER_MANAGED_INSTANCE</code>). \n Logical operators are
<code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n \n \n <p>\n <code>resourceAccount</code> - The ID of the account that owns the resource. \n Logical operators
are <code>EQUALS</code> | <code>NOT_EQUALS</code>.</p>\n \n "
 }
 }
 }
}

```

```

 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#ListUsageForLicenseConfigurationResponse": {
 "type": "structure",
 "members": {
 "LicenseConfigurationUsageList": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurationUsageList",
 "traits": {
 "smithy.api#documentation": "<p>Information about the license configurations.</p>"
 }
 },
 "NextToken": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token for the next set of results.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#Location": {
 "type": "string"
 },
 "com.amazonaws.licensemanager#Long": {
 "type": "long"
 },
 "com.amazonaws.licensemanager#ManagedResourceSummary": {
 "type": "structure",
 "members": {
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Type of resource associated with a license.</p>"
 }
 },
 "AssociationCount": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>Number of resources associated with licenses.</p>"
 }
 }
 }
 }
}

```

```

 },
 "traits": {
 "smithy.api#documentation": "<p>Summary information about a managed resource.</p>"
 }
 },
 "com.amazonaws.licensemanager#ManagedResourceSummaryList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ManagedResourceSummary"
 }
 },
 "com.amazonaws.licensemanager#MaxSize100": {
 "type": "integer",
 "traits": {
 "smithy.api#range": {
 "min": 1,
 "max": 100
 }
 }
 },
 "com.amazonaws.licensemanager#MaxSize3StringList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#String"
 },
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 3
 }
 }
 },
 "com.amazonaws.licensemanager#Message": {
 "type": "string"
 },
 "com.amazonaws.licensemanager#Metadata": {
 "type": "structure",
 "members": {
 "Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The key name.</p>"
 }
 }
 }
 },
 "Value": {
 "target": "com.amazonaws.licensemanager#String",

```

```

 "traits": {
 "smithy.api#documentation": "<p>The value.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes key/value pairs.</p>"
 }
},
"com.amazonaws.licensemanager#MetadataList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Metadata"
 }
},
"com.amazonaws.licensemanager#NoEntitlementsAllowedException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 }
},
"traits": {
 "smithy.api#documentation": "<p>There are no entitlements found for this license, or the entitlement
maximum count is reached.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
}
},
"com.amazonaws.licensemanager#Options": {
 "type": "structure",
 "members": {
 "ActivationOverrideBehavior": {
 "target": "com.amazonaws.licensemanager#ActivationOverrideBehavior",
 "traits": {
 "smithy.api#documentation": "<p>An activation option for your grant that determines the behavior of
activating a grant.\n Activation options can only be used with granted licenses sourced from the Amazon Web
Services Marketplace. Additionally, the operation must specify the value of <code>ACTIVE</code> for the\n
<code>Status</code> parameter.</p>\n \n \n <p>As a license administrator, you can
optionally specify an\n <code>ActivationOverrideBehavior</code> when activating a grant.</p>\n
\n \n <p>As a grantor, you can optionally specify an\n
<code>ActivationOverrideBehavior</code> when you activate a grant for a grantee\n account in your
organization.</p>\n \n \n <p>As a grantee, if the grantor creating the distributed
grant doesnt specify an\n <code>ActivationOverrideBehavior</code>, you can optionally specify one
when you\n are activating the grant.</p>\n \n \n <dl>\n
<dt>DISTRIBUTED_GRANTS_ONLY</dt>\n <dd>\n <p>Use this value to activate a grant
without replacing any member accounts\n active grants for the same product.</p>\n </dd>\n
<dt>ALL_GRANTS_PERMITTED_BY_ISSUER</dt>\n <dd>\n <p>Use this value to activate a

```

```

grant and disable other active grants in any\n member accounts
for the same product. This action will also replace their\n previously activated grants with this activated
grant.</p>\n </dd>\n </dl>"
 }
 }
},
"traits": {
 "smithy.api#documentation": "<p>The options you can specify when you create a new version of a grant,
such as activation\n override behavior. For more information, see Granted licenses in
License Manager in the <i>License Manager User Guide</i>.</p>"
 }
},
"com.amazonaws.licensemanager#OrganizationConfiguration": {
 "type": "structure",
 "members": {
 "EnableIntegration": {
 "target": "com.amazonaws.licensemanager#Boolean",
 "traits": {
 "smithy.api#default": false,
 "smithy.api#documentation":
"<p>Enables Organizations integration.</p>",
 "smithy.api#required": {}
 }
 }
 }
},
"traits": {
 "smithy.api#documentation": "<p>Configuration information for Organizations.</p>"
}
},
"com.amazonaws.licensemanager#PrincipalArnList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Arn"
 },
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max": 1
 }
 }
},
"com.amazonaws.licensemanager#ProductInformation": {
 "type": "structure",
 "members": {
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Resource type. The
possible values are <code>SSM_MANAGED</code> | <code>RDS</code>.</p>",
 "smithy.api#required": {}
 }
},
"ProductInformationFilterList": {
 "target": "com.amazonaws.licensemanager#ProductInformationFilterList",
 "traits": {
 "smithy.api#documentation": "<p>A Product information filter consists of a
<code>ProductInformationFilterComparator</code> which is a logical operator, a
<code>ProductInformationFilterName</code> which specifies the type of filter being declared, and a
<code>ProductInformationFilterValue</code> that specifies the value to filter on. </p>\n <p>Accepted values
for <code>ProductInformationFilterName</code> are listed here along with descriptions and valid options for
<code>ProductInformationFilterComparator</code>.</p>\n <p>The following filters and are supported when
the resource type \n is <code>SSM_MANAGED</code>:</p>\n
 \n \n <p>\n <code>Application Name</code> - The name of the
application.\n Logical operator is <code>EQUALS</code>.</p>\n \n \n <p>\n <code>Application Publisher</code> - The publisher of the application.\n Logical
operator is <code>EQUALS</code>.</p>\n \n \n <p>\n <code>Application Version</code> - The version of the application.\n Logical operator is
<code>EQUALS</code>.</p>\n \n \n <p>\n <code>Platform
Name</code> - The name of the platform.\n Logical operator is <code>EQUALS</code>.</p>\n \n \n <p>\n <code>Platform Type</code> - The platform type.\n Logical operator is <code>EQUALS</code>.</p>\n \n \n <p>\n <code>Tag:key</code> - The key of a tag attached to an
Amazon Web Services resource you wish to exclude from automated discovery. Logical operator is
<code>NOT_EQUALS</code>. The key for your tag must be appended to <code>Tag:</code> following the
example: <code>Tag:name-of-your-key</code>. <code>ProductInformationFilterValue</code> is optional if you
are not using values for the key.\n </p>\n \n \n <p>\n <code>AccountId</code> - The 12-digit ID of an Amazon Web Services account you wish to exclude from
automated discovery.\n Logical operator is <code>NOT_EQUALS</code>.</p>\n \n \n <p>\n <code>License Included</code> - The type of license included.\n Logical
operators are <code>EQUALS</code> and <code>NOT_EQUALS</code>.\n Possible values are:
<code>sql-server-enterprise</code> | \n <code>sql-server-standard</code> | \n <code>sql-server-
web</code> | \n <code>windows-server-datacenter</code>.</p>\n \n \n <p>The following filters and logical operators are supported when the resource type\n is
<code>RDS</code>:</p>\n \n \n <p>\n <code>Engine Edition</code> -
The edition of the database engine.\n Logical operator is <code>EQUALS</code>.\n Possible
values are: <code>oracle-ee</code> | <code>oracle-se</code> | <code>oracle-se1</code> | <code>oracle-
se2</code>.</p>\n \n \n <p>\n <code>License Pack</code> - The license
pack.\n Logical operator is <code>EQUALS</code>.\n Possible values are: <code>data
guard</code> | \n <code>diagnostic pack sqlt</code> | \n <code>tuning pack sqlt</code> | \n <code>ols</code> | \n <code>olap</code>.</p>\n \n \n "smithy.api#required": {}
 }
}
},

```

```

 "traits": {
 "smithy.api#documentation": "<p>Describes product information for a license configuration.</p>"
 }
 },
 "com.amazonaws.licensemanager#ProductInformationFilter": {
 "type": "structure",
 "members": {
 "ProductInformationFilterName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Filter name.</p>",
 "smithy.api#required": {}
 }
 },
 "ProductInformationFilterValue": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>Filter value.</p>"
 }
 },
 "ProductInformationFilterComparator": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Logical operator.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Describes product information filters.</p>"
 }
 },
 "com.amazonaws.licensemanager#ProductInformationFilterList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ProductInformationFilter"
 }
 },
 "com.amazonaws.licensemanager#ProductInformationList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ProductInformation"
 }
 },
 "com.amazonaws.licensemanager#ProvisionalConfiguration": {
 "type": "structure",
 "members": {

```

```

 "MaxTimeToLiveInMinutes": {
 "target": "com.amazonaws.licensemanager#BoxInteger",
 "traits": {
 "smithy.api#documentation": "<p>Maximum time for the provisional configuration, in minutes.</p>",
 "smithy.api#required": {}
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about a provisional configuration.</p>"
 }
 },
 "com.amazonaws.licensemanager#RateLimitExceededException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "RateLimitExceeded",
 "httpResponseCode": 429
 },
 "smithy.api#documentation": "<p>Too many requests have been submitted. Try again after a brief
wait.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 429
 }
 },
 "com.amazonaws.licensemanager#ReceivedMetadata": {
 "type": "structure",
 "members": {
 "ReceivedStatus": {
 "target": "com.amazonaws.licensemanager#ReceivedStatus",
 "traits": {
 "smithy.api#documentation": "<p>Received status.</p>"
 }
 },
 "ReceivedStatusReason": {
 "target": "com.amazonaws.licensemanager#StatusReasonMessage",
 "traits": {
 "smithy.api#documentation": "<p>Received
status reason.</p>"
 }
 }
 },
 "AllowedOperations": {
 "target": "com.amazonaws.licensemanager#AllowedOperationList",

```



```

 "traits": {
 "smithy.api#documentation": "<p>Allowed operations.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Metadata associated with received licenses and grants.</p>"
 }
},
"com.amazonaws.licensemanager#ReceivedStatus": {
 "type": "enum",
 "members": {
 "PENDING_WORKFLOW": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "PENDING_WORKFLOW"
 }
 },
 "PENDING_ACCEPT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue":
"PENDING_ACCEPT"
 }
 },
 "REJECTED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "REJECTED"
 }
 },
 "ACTIVE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "ACTIVE"
 }
 },
 "FAILED_WORKFLOW": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "FAILED_WORKFLOW"
 }
 },
 "DELETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DELETED"
 }
 }
 }
}

```

```

 },
 "DISABLED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DISABLED"
 }
 },
 "WORKFLOW_COMPLETED": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "WORKFLOW_COMPLETED"
 }
 }
 }
},
"com.amazonaws.licensemanager#RedirectException": {
 "type": "structure",
 "members": {
 "Location": {
 "target": "com.amazonaws.licensemanager#Location",
 "traits": {
 "smithy.api#httpHeader": "Location"
 }
 },
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>This is not the correct Region for the resource. Try again.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 308
 }
},
"com.amazonaws.licensemanager#RejectGrant": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#RejectGrantRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#RejectGrantResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 }
]
}

```

```

 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target":
"com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Rejects the specified grant.</p>"
 }
},
"com.amazonaws.licensemanager#RejectGrantRequest": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the grant.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
}
},
"com.amazonaws.licensemanager#RejectGrantResponse": {
 "type": "structure",
 "members": {
 "GrantArn": {
 "target": "com.amazonaws.licensemanager#Arn",
 "traits": {
 "smithy.api#documentation": "<p>Grant ARN.</p>"
 }
 }
 },
 "Status": {

```

```

 "target": "com.amazonaws.licensemanager#GrantStatus",
 "traits": {
 "smithy.api#documentation": "<p>Grant status.</p>"
 }
 },
 "Version": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Grant version.</p>"
 }
 }
},
"traits": {
 "smithy.api#output": {}
}
},
"com.amazonaws.licensemanager#RenewType":
{
 "type": "enum",
 "members": {
 "NONE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "None"
 }
 },
 "WEEKLY": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Weekly"
 }
 },
 "MONTHLY": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "Monthly"
 }
 }
 }
},
"com.amazonaws.licensemanager#ReportContext": {
 "type": "structure",
 "members": {
 "licenseConfigurationArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {

```

```

 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration that this
generator reports on.</p>",
 "smithy.api#required": {}
 }
},
"traits": {
 "smithy.api#documentation": "<p>Details of the license configuration that this generator reports on.</p>"
}
},
"com.amazonaws.licensemanager#ReportFrequency": {
 "type": "structure",
 "members": {
 "value": {
 "target": "com.amazonaws.licensemanager#Integer",
 "traits": {
 "smithy.api#documentation": "Number of times within the frequency period that a report is
generated. \n The only supported value is <code>1</code>.</p>"
 }
 },
 "period": {
 "target": "com.amazonaws.licensemanager#ReportFrequencyType",

 "traits": {
 "smithy.api#documentation": "<p>Time period between each report. The period can be daily, weekly,
or monthly.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
 }
},
"com.amazonaws.licensemanager#ReportFrequencyType": {
 "type": "enum",
 "members": {
 "DAY": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "DAY"
 }
 },
 "WEEK": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "WEEK"
 }
 }
 },
}

```

```

 "MONTH": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "MONTH"
 }
 }
 },
 "com.amazonaws.licensemanager#ReportGenerator": {
 "type": "structure",
 "members": {
 "ReportGeneratorName": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the report generator.</p>"
 }
 },
 "ReportType": {
 "target": "com.amazonaws.licensemanager#ReportTypeList",
 "traits": {
 "smithy.api#documentation": "<p>Type of reports that are generated.</p>"
 }
 },
 "ReportContext": {
 "target": "com.amazonaws.licensemanager#ReportContext",
 "traits": {
 "smithy.api#documentation": "<p>License
configuration type for this generator.</p>"
 }
 },
 "ReportFrequency": {
 "target": "com.amazonaws.licensemanager#ReportFrequency",
 "traits": {
 "smithy.api#documentation": "<p>Details about how frequently reports are generated.</p>"
 }
 },
 "LicenseManagerReportGeneratorArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator.</p>"
 }
 },
 "LastRunStatus": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Status of the last report generation attempt.</p>"
 }
 }
 }
 }
}

```

```

 },
 "LastRunFailureReason":
 {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Failure message for the last report generation attempt.</p>"
 }
 },
 "LastReportGenerationTime": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Time the last report was generated at.</p>"
 }
 },
 "ReportCreatorAccount": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>The Amazon Web Services account ID used to create the report
generator.</p>"
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",

 "traits": {
 "smithy.api#documentation": "<p>Description of the report generator.</p>"
 }
 },
 "S3Location": {
 "target": "com.amazonaws.licensemanager#S3Location",
 "traits": {
 "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published
to.</p>"
 }
 },
 "CreateTime": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Time the report was created.</p>"
 }
 },
 "Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>Tags associated with the report generator.</p>"
 }
 }
 }
}

```

```

 },
 "traits": {
 "smithy.api#documentation": "<p>Describe the details of a report generator.</p>"
 }
 },
 "com.amazonaws.licensemanager#ReportGeneratorList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ReportGenerator"
 }
 },
 "com.amazonaws.licensemanager#ReportGeneratorName": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 1,
 "max": 100
 }
 }
 },
 "com.amazonaws.licensemanager#ReportType": {
 "type": "enum",
 "members": {
 "LICENSE_CONFIGURATION_SUMMARY_REPORT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "LicenseConfigurationSummaryReport"
 }
 },
 "LICENSE_CONFIGURATION_USAGE_REPORT": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "LicenseConfigurationUsageReport"
 }
 }
 }
 },
 "com.amazonaws.licensemanager#ReportTypeList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ReportType"
 }
 },
 "com.amazonaws.licensemanager#ResourceInventory": {
 "type": "structure",
 "members": {

```



```

 "ResourceId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>ID of the resource.</p>"
 }
 },
 "ResourceType": {
 "target": "com.amazonaws.licensemanager#ResourceType",
 "traits": {
 "smithy.api#documentation": "<p>Type of resource.</p>"
 }
 },
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the resource.</p>"
 }
 },
 "Platform": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Platform of the resource.</p>"
 }
 },
 "PlatformVersion": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Platform version of the resource in the inventory.</p>"
 }
 },
 "ResourceOwningAccountId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>ID of the account that owns the resource.</p>"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details about a resource.</p>"
 },
 "com.amazonaws.licensemanager#ResourceInventoryList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#ResourceInventory"
 }
 }
 }

```

```

},
"com.amazonaws.licensemanager#ResourceLimitExceededException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "ResourceLimitExceeded",

 "httpResponseCode": 400
 },
 "smithy.api#documentation": "<p>Your resource limits have been exceeded.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
},
"com.amazonaws.licensemanager#ResourceNotFoundException": {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "aws.protocols#awsQueryError": {
 "code": "InvalidResource.NotFound",
 "httpResponseCode": 400
 },
 "smithy.api#documentation": "<p>The resource cannot be found.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
},
"com.amazonaws.licensemanager#ResourceType": {
 "type": "enum",
 "members": {
 "EC2_INSTANCE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "EC2_INSTANCE"
 }
 },
 "EC2_HOST": {
 "target": "smithy.api#Unit",
 "traits": {

```

```

 "smithy.api#enumValue": "EC2_HOST"
 },
 "EC2_AMI": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "EC2_AMI"
 }
 },
 "RDS": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "RDS"
 }
 },
 "SYSTEMS_MANAGER_MANAGED_INSTANCE": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "SYSTEMS_MANAGER_MANAGED_INSTANCE"
 }
 }
},
"com.amazonaws.licensemanager#S3Location": {
 "type": "structure",
 "members": {
 "bucket": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Name of the S3 bucket reports are published to.</p>"
 }
 },
 "keyPrefix": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Prefix of the S3 bucket reports are published to.</p>"
 }
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>Details of the S3 bucket that report generator reports are published to.</p>"
 }
},
"com.amazonaws.licensemanager#ServerInternalException": {
 "type": "structure",
 "members": {
 "Message": {

```

```

 "target": "com.amazonaws.licensemanager#Message"
 }
},
"traits": {
 "aws.protocols#awsQueryError": {
 "code": "InternalServerError",
 "httpResponseCode": 500
 },
 "smithy.api#documentation": "<p>The server experienced an internal error. Try again.</p>",
 "smithy.api#error": "server",
 "smithy.api#httpError": 500
}
},
"com.amazonaws.licensemanager#SignedToken": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 4096
 }
 }
},
"com.amazonaws.licensemanager#StatusReasonMessage": {
 "type": "string",
 "traits":
{
 "smithy.api#length": {
 "min": 0,
 "max": 400
 },
 "smithy.api#pattern": "^[\\s\\S]+$"
 }
},
"com.amazonaws.licensemanager#String": {
 "type": "string"
},
"com.amazonaws.licensemanager#StringList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#String"
 }
},
"com.amazonaws.licensemanager#Tag": {
 "type": "structure",
 "members": {
 "Key": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Tag key.</p>"
 }
 }
 }
}

```

```

 }
 },
 "Value": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation":
"<p>Tag value.</p>"
 }
 }
},
"traits": {
 "smithy.api#documentation": "<p>Details about a tag for a license configuration.</p>"
}
},
"com.amazonaws.licensemanager#TagKeyList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#String"
 }
},
"com.amazonaws.licensemanager#TagList": {
 "type": "list",
 "member": {
 "target": "com.amazonaws.licensemanager#Tag"
 }
},
"com.amazonaws.licensemanager#TagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#TagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#TagResourceResponse"
 },
 "errors": [
 {
 "target":
"com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 }
]
}

```

```

 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Adds the specified tags to the specified license configuration.</p>"
 }
},
"com.amazonaws.licensemanager#TagResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
 "smithy.api#required": {}
 }
 },
 "Tags": {
 "target": "com.amazonaws.licensemanager#TagList",
 "traits": {
 "smithy.api#documentation": "<p>One or more tags.</p>",
 "smithy.api#required": {}
 }
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#TagResourceResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#TokenData": {
 "type": "structure",
 "members": {
 "TokenId": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token ID.</p>"
 }
 }
 },
 },

```

```

"TokenType": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Type of token generated. The supported value is
<code>REFRESH_TOKEN</code>.</p>"
 }
},
"LicenseArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license.</p>"
 }
},
"ExpirationTime": {
 "target": "com.amazonaws.licensemanager#ISO8601DateTime",
 "traits": {
 "smithy.api#documentation": "<p>Token expiration time, in ISO8601-UTC
format.</p>"
 }
},
"TokenProperties": {
 "target": "com.amazonaws.licensemanager#MaxSize3StringList",
 "traits": {
 "smithy.api#documentation": "<p>Data specified by the caller.</p>"
 }
},
"RoleArns": {
 "target": "com.amazonaws.licensemanager#ArnList",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Names (ARN) of the roles included in the
token.</p>"
 }
},
"Status": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Token status. The possible values are
<code>AVAILABLE</code> and <code>DELETED</code>.</p>"
 }
},
"traits": {
 "smithy.api#documentation": "<p>Describes a token.</p>"
}
},
"com.amazonaws.licensemanager#TokenList": {
 "type": "list",

```

```

 "member": {
 "target": "com.amazonaws.licensemanager#TokenData"
 }
 },
 "com.amazonaws.licensemanager#TokenString": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 4096
 },
 "smithy.api#pattern": "^\\S+$"
 }
 },
 "com.amazonaws.licensemanager#TokenType": {
 "type": "enum",
 "members": {
 "REFRESH_TOKEN": {
 "target": "smithy.api#Unit",
 "traits": {
 "smithy.api#enumValue": "REFRESH_TOKEN"
 }
 }
 }
 },
 "com.amazonaws.licensemanager#UnsupportedDigitalSignatureMethodException":
 {
 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The digital signature method is unsupported. Try your request again.</p>",
 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
 },
 "com.amazonaws.licensemanager#UntagResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#UntagResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#UntagResourceResponse"
 }
 },

```



```

"errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
"traits": {
 "smithy.api#documentation": "<p>Removes the specified tags from the specified license
configuration.</p>"
},
},
"com.amazonaws.licensemanager#UntagResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license
configuration.</p>",
 "smithy.api#required": {}
 }
 },
 },
 "TagKeys": {
 "target": "com.amazonaws.licensemanager#TagKeyList",
 "traits": {
 "smithy.api#documentation": "<p>Keys identifying the tags to remove.</p>",
 "smithy.api#required": {}
 }
 },
},
"traits": {
 "smithy.api#input": {}
},
},
"com.amazonaws.licensemanager#UntagResourceResponse": {
 "type": "structure",
 "members": {},

```

```

 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#UpdateLicenseConfiguration": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Modifies the attributes of an existing license configuration.</p>"
 }
 },
 "com.amazonaws.licensemanager#UpdateLicenseConfigurationRequest": {
 "type": "structure",
 "members": {
 "LicenseConfigurationArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the license configuration.</p>",
 "smithy.api#required": {}
 }
 },
 },
 },

```

```

"LicenseConfigurationStatus": {
 "target": "com.amazonaws.licensemanager#LicenseConfigurationStatus",
 "traits": {
 "smithy.api#documentation": "<p>New status of the license configuration.</p>"
 }
},
"LicenseRules": {
 "target": "com.amazonaws.licensemanager#StringList",
 "traits": {
 "smithy.api#documentation": "<p>New license rule. The only rule that you can add after you create a license\nconfiguration is licenseAffinityToHost.</p>"
 }
},
"LicenseCount": {
 "target": "com.amazonaws.licensemanager#BoxLong",
 "traits": {
 "smithy.api#documentation": "<p>New number of licenses managed by the license\nconfiguration.</p>"
 }
},
"LicenseCountHardLimit": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>New hard limit of the number of available licenses.</p>"
 }
},
"Name": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>New name\nof the license configuration.</p>"
 }
},
"Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>New description of the license configuration.</p>"
 }
},
"ProductInformationList": {
 "target": "com.amazonaws.licensemanager#ProductInformationList",
 "traits": {
 "smithy.api#documentation": "<p>New product information.</p>"
 }
},
"DisassociateWhenNotFound": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",

```

```

 "traits": {
 "smithy.api#documentation": "<p>When true, disassociates a resource when software is
uninstalled.</p>"
 }
 },
 "traits": {
 {
 "smithy.api#input": {}
 }
 },
 "com.amazonaws.licensemanager#UpdateLicenseConfigurationResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
 },
 "com.amazonaws.licensemanager#UpdateLicenseManagerReportGenerator": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ResourceNotFoundException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 },
],
 },

```

```

 {
 "target": "com.amazonaws.licensemanager#ValidationException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Updates a report generator.</p>\n <p>After you make changes to a
report generator, it starts generating new reports within 60 minutes of being updated.</p>"
 }
},
"com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorRequest":
{
 "type": "structure",
 "members": {
 "LicenseManagerReportGeneratorArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the report generator to
update.</p>",
 "smithy.api#required": {}
 }
 },
 "ReportGeneratorName": {
 "target": "com.amazonaws.licensemanager#ReportGeneratorName",
 "traits": {
 "smithy.api#documentation": "<p>Name of the report generator.</p>",
 "smithy.api#required": {}
 }
 },
 "Type": {
 "target": "com.amazonaws.licensemanager#ReportTypeList",
 "traits": {
 "smithy.api#documentation":
 "<p>Type of reports to generate. The following report types are supported:</p>\n \n \n <p>License configuration report - Reports the number and details of consumed licenses for a license
configuration.</p>\n \n \n <p>Resource report - Reports the tracked licenses and
resource consumption for a license configuration.</p>\n \n ",
 "smithy.api#required": {}
 }
 },
 "ReportContext": {
 "target": "com.amazonaws.licensemanager#ReportContext",
 "traits": {
 "smithy.api#documentation": "<p>The report context.</p>",
 "smithy.api#required": {}
 }
 },
 "ReportFrequency": {
 "target": "com.amazonaws.licensemanager#ReportFrequency",

```

```

 "traits":
 {
 "smithy.api#documentation": "<p>Frequency by which reports are generated.</p>",
 "smithy.api#required": {}
 },
 "ClientToken": {
 "target": "com.amazonaws.licensemanager#ClientRequestToken",
 "traits": {
 "smithy.api#documentation": "<p>Unique, case-sensitive identifier that you provide to ensure the
idempotency of the request.</p>",
 "smithy.api#required": {}
 }
 },
 "Description": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Description of the report generator.</p>"
 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#UpdateLicenseManagerReportGeneratorResponse":
{
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResource": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 }
],

```

```

 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },

 {
 "target": "com.amazonaws.licensemanager#InvalidResourceStateException"
 },
 {
 "target": "com.amazonaws.licensemanager#LicenseUsageException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Adds or removes the specified license configurations for the specified Amazon Web Services resource.</p>\n <p>You can update the license specifications of AMIs, instances, and hosts.\n You cannot update the license specifications for launch templates and CloudFormation templates, \n as they send license configurations to the operation that creates the resource.</p>"
 }
},

"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceRequest": {
 "type": "structure",
 "members": {
 "ResourceArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon Web Services resource.</p>",
 "smithy.api#required": {}
 }
 },
 "AddLicenseSpecifications": {
 "target": "com.amazonaws.licensemanager#LicenseSpecifications",
 "traits": {
 "smithy.api#documentation": "<p>ARNs of the license configurations to add.</p>"
 }
 },
 "RemoveLicenseSpecifications": {
 "target": "com.amazonaws.licensemanager#LicenseSpecifications",
 "traits": {
 "smithy.api#documentation":
 "<p>ARNs of the license configurations to remove.</p>"
 }
 }
 }
}

```

```

 }
 },
 "traits": {
 "smithy.api#input": {}
 }
},
"com.amazonaws.licensemanager#UpdateLicenseSpecificationsForResourceResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#UpdateServiceSettings": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanager#UpdateServiceSettingsRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanager#UpdateServiceSettingsResponse"
 },
 "errors": [
 {
 "target": "com.amazonaws.licensemanager#AccessDeniedException"
 },
 {
 "target": "com.amazonaws.licensemanager#AuthorizationException"
 },
 {
 "target": "com.amazonaws.licensemanager#InvalidParameterValueException"
 },
 {
 "target": "com.amazonaws.licensemanager#RateLimitExceededException"
 },
 {
 "target": "com.amazonaws.licensemanager#ServerInternalException"
 }
],
 "traits": {
 "smithy.api#documentation": "<p>Updates License Manager settings for the current Region.</p>"
 }
},
"com.amazonaws.licensemanager#UpdateServiceSettingsRequest": {
 "type": "structure",
 "members": {
 "S3BucketArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {

```



```

 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon S3 bucket where
the License
Manager information is stored.</p>"
 },
 "SnsTopicArn": {
 "target": "com.amazonaws.licensemanager#String",
 "traits": {
 "smithy.api#documentation": "<p>Amazon Resource Name (ARN) of the Amazon SNS topic used
for License Manager alerts.</p>"
 }
 },
 "OrganizationConfiguration": {
 "target": "com.amazonaws.licensemanager#OrganizationConfiguration",
 "traits": {
 "smithy.api#documentation": "<p>Enables integration with Organizations for cross-account
discovery.</p>"
 }
 },
 "EnableCrossAccountsDiscovery": {
 "target": "com.amazonaws.licensemanager#BoxBoolean",
 "traits": {
 "smithy.api#documentation": "<p>Activates cross-account discovery.</p>"
 }
 }
},
"traits": {
 "smithy.api#input": {}
}
},
"com.amazonaws.licensemanager#UpdateServiceSettingsResponse": {
 "type": "structure",
 "members": {},
 "traits": {
 "smithy.api#output": {}
 }
},
"com.amazonaws.licensemanager#UsageOperation": {
 "type": "string",
 "traits": {
 "smithy.api#length": {
 "min": 0,
 "max": 50
 }
 }
},
"com.amazonaws.licensemanager#ValidationException": {

```

```

 "type": "structure",
 "members": {
 "Message": {
 "target": "com.amazonaws.licensemanager#Message"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",

 "smithy.api#error": "client",
 "smithy.api#httpError": 400
 }
}
}
}

```

CheckoutLicense

Initialize stack step

spanInitializeStart

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

Retry

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata  
ErrorCloseResponseBody  
CloseResponseBody  
ResponseErrorWrapper  
RequestIDRetriever  
OperationDeserializer  
AddTimeOffsetMiddleware  
RecordResponseTiming  
RequestResponseLogger

## 1.89 procfs 0.16.1

### 1.89.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at  
SoundCloud Ltd. (<http://soundcloud.com/>).

# 1.90 github.com/confluentinc/confluent-kafka-go/v2 2.10.0

## 1.90.1 Available under license :

LICENSE

-----  
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill  
2023, Confluent Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,  
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimer in the documentation  
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE  
LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

-----  
For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished  
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE



AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

-----  
# For src/crc32c.c copied (with modifications) from  
# <http://stackoverflow.com/a/17646775/1821055>

/\* crc32c.c -- compute CRC-32C using the Intel crc32 instruction  
\* Copyright (C) 2013 Mark Adler  
\* Version 1.1 1 Aug 2013 Mark Adler  
\*/

/\*  
This software is provided 'as-is',  
without any express or implied  
warranty. In no event will the author be held liable for any damages  
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,  
including commercial applications, and to alter it and redistribute it  
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler  
madler@alumni.caltech.edu  
\*/

LICENSE.fnv1a

-----  
parts of src/rdfnv1a.c: [http://www.isthe.com/chongo/src/fnv/hash\\_32a.c](http://www.isthe.com/chongo/src/fnv/hash_32a.c)

Please  
do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> ^oo^  
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

-----  
This license covers src/rdhdrhistogram.c which is a C port of  
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>  
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

-----  
The MIT License  
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

LICENSE.lz4

-----  
src/rdxxhash.[ch] src/lz4\*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this  
list of conditions and the following disclaimer in the documentation and/or  
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  
PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

-----  
parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

//-----  
// MurmurHash2 was written by Austin Appleby, and is placed in the public  
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person  
obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

-----  
For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.opentelemetry

-----  
For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution

incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE  
text from the Work, provided  
that such additional attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions  
for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding  
such Contributions.

6. Trademarks. This License does not grant permission to use the trade  
names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the



same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

LICENSE.pycrc

-----  
The following license applies to the files rdcrc32.c and rdcrc32.h which  
have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,

free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

LICENSE.queue

For

sys/queue.h:

```
* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
* ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$
```

LICENSE.regexp

-----  
regexp.c and regexp.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

-----  
#####

```
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219
#####
```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinycthread

-----  
From <https://github.com/tinycthread/tinycthread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

-----  
Copyright (c) 2012 Marcus Geelnard  
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

-----  
For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

/\*

\* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

\*

\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
\* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR  
\* ANY DAMAGES

\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
\* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
\* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

\*

\* Sponsored in part by the Defense Advanced Research Projects  
\* Agency (DARPA) and Air Force Research Laboratory, Air Force  
\* Materiel Command, USAF, under agreement number F39502-99-1-0512.

\*/

/\*-

\* Copyright (c) 2000 The NetBSD Foundation, Inc.

\* All rights reserved.

\*  
 \* This code is derived from software contributed to The NetBSD Foundation  
 \* by Dieter Baron and Thomas Klausner.  
 \*  
 \* Redistribution and use in source and binary forms, with or without  
 \* modification, are permitted provided that the following conditions  
 \* are met:  
 \* 1. Redistributions of source code must retain the above copyright  
 \* notice, this list of conditions and the following disclaimer.  
 \* 2. Redistributions in binary form must reproduce the above copyright  
 \* notice,  
 this list of conditions and the following disclaimer in the  
 \* documentation and/or other materials provided with the distribution.  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS  
 \* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED  
 \* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
 \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS  
 \* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
 \* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
 \* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
 \* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
 \* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
 \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
 \* POSSIBILITY OF SUCH DAMAGE.  
 \*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
 and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
 the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
 other entities that control, are controlled by, or are under common  
 control with that entity. For the purposes of this definition,  
 "control" means (i) the power, direct or indirect, to cause the  
 direction or management of such entity, whether by contract or  
 otherwise, or (ii) ownership of fifty percent (50%) or more of the  
 outstanding shares, or (iii) beneficial ownership of such entity.

"You"  
(or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but  
not limited to compiled object code, generated documentation,  
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or  
Object form, made available under the License, as indicated by a  
copyright notice that is included in or attached to the work  
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object  
form, that is based on (or derived from) the Work and for which the  
editorial revisions, annotations, elaborations,  
or other modifications  
represent, as a whole, an original work of authorship. For the purposes  
of this License, Derivative Works shall not include works that remain  
separable from, or merely link (or bind by name) to the interfaces of,  
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including  
the original version of the Work and any modifications or additions  
to that Work or Derivative Works thereof, that is intentionally  
submitted to Licensor for inclusion in the Work by the copyright owner  
or by an individual or Legal Entity authorized to submit on behalf of  
the copyright owner. For the purposes of this definition, "submitted"  
means any form of electronic, verbal, or written communication sent  
to the Licensor or its representatives, including but not limited to  
communication on electronic mailing lists, source code control systems,  
and issue tracking systems  
that are managed by, or on behalf of, the  
Licensor for the purpose of discussing and improving the Work, but  
excluding communication that is conspicuously marked or otherwise  
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity  
on behalf of whom a Contribution has been received by Licensor and  
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of  
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,



or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.91 golang 1.24.3

## 1.91.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit  
(<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

## 1.91.2 Available under license :

// Copyright 2024 The Go Authors. All rights reserved.

// Use of this source code is governed by a BSD-style

// license that can be found in the LICENSE file.

package copyright

```
import (
 "bytes"
 "internal/testenv"
 "io"
 "io/fs"
 "os"
 "path/filepath"
 "testing"
)
```

```
var copyright = []byte("Copyright")
```

```
var permitted = [][]byte{
 []byte("// Code generated by "),
 []byte("// Code generated from "),
 []byte("// Created by cgo -cdefs"),
 []byte("// DO NOT EDIT\n// generated by:"),
 []byte("// Empty assembly file"),
 []byte("// Generated using cgo"),
 []byte("// Original source:\n//\nhttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
 []byte("// created by cgo -cdefs"),
 []byte("// go run mkasm.go"),
 []byte("// mkerrors"),
 []byte("// mksys"),
 []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}
```

```
func TestCopyright(t *testing.T) {
 buf := make([]byte, 2048)
 filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
 d fs.DirEntry, err error) error {
 if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
 return filepath.SkipDir
 }
 })
}
```

```

}
switch filepath.Ext(d.Name()) {
default:
 return nil
case ".s", ".go":
 // check
}

f, err := os.Open(path)
if err != nil {
 t.Error(err)
 return nil
}
defer f.Close()
n, err := f.Read(buf)
if err != nil && err != io.EOF {
 t.Error(err)
 return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
 return nil
}
for _, ok := range permitted {
 if bytes.HasPrefix(b, ok) {
 return nil
 }
}
t.Errorf("%s: missing copyright notice", path)
return nil
})
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto\_linux\_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at [https://boringssl.googlesource.com/boringssl/+/\\_fips-20220613/LICENSE](https://boringssl.googlesource.com/boringssl/+/_fips-20220613/LICENSE).

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: <https://cla.developers.google.com/cla>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

-----

```

/* =====
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following
disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit
(http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED

```

\* OF THE POSSIBILITY OF SUCH DAMAGE.

\* =====

\*

\* This product includes cryptographic software written by Eric Young

\* (eay@cryptsoft.com). This product

includes software written by Tim

\* Hudson (tjh@cryptsoft.com).

\*

\*/

Original SSLeay License

-----

/\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

\* All rights reserved.

\*

\* This package is an SSL implementation written

\* by Eric Young (eay@cryptsoft.com).

\* The implementation was written so as to conform with Netscapes SSL.

\*

\* This library is free for commercial and non-commercial use as long as

\* the following conditions are aheared to. The following conditions

\* apply to all code found in this distribution, be it the RC4, RSA,

\* lhash, DES, etc., code; not just the SSL code. The SSL documentation

\* included with this distribution is covered by the same copyright terms

\* except that the holder is Tim Hudson (tjh@cryptsoft.com).

\*

\* Copyright remains Eric Young's, and as such any Copyright notices in

\* the code are not to be removed.

\* If this package is used in a product, Eric Young should be given attribution

\* as the author of

the parts of the library used.

\* This can be in the form of a textual message at program startup or

\* in documentation (online or textual) provided with the package.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* "This product includes cryptographic software written by

\* Eric Young (eay@cryptsoft.com)"

\* The word 'cryptographic' can be left out if the rouines from the library

```

* being used are
not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

```

ISC license used for completely new code in BoringSSL:

```

/* Copyright (c) 2015, Google Inc.
*
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

```

Some files from Intel carry the following license:

```

Copyright (c) 2012, Intel Corporation
#

```

```
All rights reserved.
#
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
#
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
#
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.
#
* Neither the name of the Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
#
#
THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
""AS IS"" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
```



Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Andrea Leofreddi.

## 1.92 x-net 0.40.0

### 1.92.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.93 x-sys 0.33.0

### 1.93.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT



LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.94 x-text 0.25.0

### 1.94.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.95 stdlib go1.24.3

## 1.95.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.
// Use of this source code is governed by a BSD-style
// license that can be found in the LICENSE file.
```

```
package copyright
```

```
import (
 "bytes"
 "internal/testenv"
 "io"
 "io/fs"
 "os"
 "path/filepath"
 "testing"
)
```

```
var copyright = []byte("Copyright")
```

```
var permitted = [][]byte{
 []byte("// Code generated by "),
 []byte("// Code generated from "),
 []byte("// Created by cgo -cdefs"),
 []byte("// DO NOT EDIT\n// generated by:"),
 []byte("// Empty assembly file"),
 []byte("// Generated using cgo"),
 []byte("// Original source:\n//\nhttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
 []byte("// created by cgo -cdefs"),
 []byte("// go run mkasm.go"),
 []byte("// mkerrors"),
 []byte("// mksys"),
 []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}
```

```
func TestCopyright(t *testing.T) {
 buf := make([]byte, 2048)
 filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
 d fs.DirEntry, err error) error {
 if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
 return filepath.SkipDir
 }
 switch filepath.Ext(d.Name()) {
 default:
 return nil
 case ".s", ".go":
 // check
 }
 })
}
```

```

f, err := os.Open(path)
if err != nil {
 t.Error(err)
 return nil
}
defer f.Close()
n, err := f.Read(buf)
if err != nil && err != io.EOF {
 t.Error(err)
 return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
 return nil
}
for _, ok := range permitted {
 if bytes.HasPrefix(b, ok) {
 return nil
 }
}
t.Errorf("%s: missing copyright notice", path)
return nil
}))
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto\_linux\_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at <https://boringssl.googlesource.com/boringssl/+fips-20220613/LICENSE>.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: <https://cla.developers.google.com/clas>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

-----

/\* =====

\* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

- \* 1. Redistributions of source code must retain the above copyright
- \* notice, this list of conditions and the following disclaimer.
- \*
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following
- disclaimer in
- \* the documentation and/or other materials provided with the
- \* distribution.
- \*
- \* 3. All advertising materials mentioning features or use of this
- \* software must display the following acknowledgment:
- \* "This product includes software developed by the OpenSSL Project
- \* for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- \*
- \* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
- \* endorse or promote products derived from this software without
- \* prior written permission. For written permission, please contact
- \* [openssl-core@openssl.org](mailto:openssl-core@openssl.org).
- \*
- \* 5. Products derived from this software may not be called "OpenSSL"
- \* nor may "OpenSSL" appear in their names without prior written
- \* permission of the OpenSSL Project.
- \*
- \* 6. Redistributions of any form whatsoever must retain the following
- \* acknowledgment:
- \* "This product includes software developed by the OpenSSL Project
- \* for use in the OpenSSL Toolkit
- (<http://www.openssl.org/>)"
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- \* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- \* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- \* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- \* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- \* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- \* OF THE POSSIBILITY OF SUCH DAMAGE.
- \* =====
- \*
- \* This product includes cryptographic software written by Eric Young
- \* ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). This product
- includes software written by Tim
- \* Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).
- \*

\*/

## Original SSLeay License

-----

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of
 * the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are
 * not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
 \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
 \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
 \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
 \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
 \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
 POSSIBILITY OF  
 \* SUCH DAMAGE.  
 \*  
 \* The licence and distribution terms for any publically available version or  
 \* derivative of this code cannot be changed. i.e. this code cannot simply be  
 \* copied and put under another distribution licence  
 \* [including the GNU Public Licence.]  
 \*/

ISC license used for completely new code in BoringSSL:

```
/* Copyright (c) 2015, Google Inc.
 *
 * Permission to use, copy, modify, and/or distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
 * SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
 * OF CONTRACT, NEGLIGENCE
 * OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
 * CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */
```

Some files from Intel carry the following license:

```
Copyright (c) 2012, Intel Corporation
#
All rights reserved.
#
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
#
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
```

```

#
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.
#
* Neither the name of the Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
#
#
THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
""AS IS"" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above



copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the

distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are

permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Andrea Leofreddi.

## 1.96 x-crypto 0.38.0

### 1.96.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This source code was written by the Go contributors.

# The master list of contributors is in the main Go distribution,

# visible at <https://tip.golang.org/CONTRIBUTORS>.

## 1.97 go-auth-oauth2adapt 0.2.8

### 1.97.1 Available under license :

// Copyright 2024 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <http://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and



```

// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
 }
 for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0

```

```
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := procurement.NewLicenseManagementRESTClient(ctx)
 if err != nil {
```

```

// TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleLicenseManagementClient_Assign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
 }
 resp, err := c.Assign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 //
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()
}

```

```

req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
dUsersRequest.
}
it := c.EnumerateLicensedUsers(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 //
 TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
}
}

func ExampleLicenseManagementClient_GetLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //
 See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRe
quest.
 }
 resp, err := c.GetLicensePool(ctx, req)

```

```

if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_Unassign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UnassignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest.
 }
 resp,
 err := c.Unassign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_UpdateLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UpdateLicensePoolRequest{

```

```

// TODO: Fill request struct fields.
// See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
}
resp, err := c.UpdateLicensePool(ctx, req)
if err != nil {
 // TODO: Handle error.
}
//
// TODO: Use resp.
_ = resp
}

```

```

func ExampleLicenseManagementClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }
 resp, err := c.GetOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,

```

```

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.DeleteLicenseRequest{

```

```

//
TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#DeleteLicenseRequest.
}
op, err := c.Delete(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options

```



```

c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
}
resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_List() {

```

```

ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template
only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
}
it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*computepb.LicensesListResponse)
}
}

func
ExampleLicensesClient_SetIamPolicy() {
ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}

```

```

}
defer c.Close()

req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#SetIamPolicyLicenseRequest.
}
resp, err := c.SetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically
 // generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.



```

// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var
 newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
 return &LicensesCallOptions{
 Delete: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 Get: []gax.CallOption{

```

```

gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
)),
},
GetIamPolicy:
[]gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
)),
},
Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
)),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

```

```

//
internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption)
(*computepb.TestPermissionsResponse,
error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.
 CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection

```

```

to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty
// if no such policy or resource exists. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that
// belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a
// list of publicly-available licenses,
// use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution
// This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This

```

```

resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners
who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicensesClient
 CallOptions **LicensesCallOptions

 logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx
context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()

```

```

c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {
 return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN")
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {

```

```

// Replace httpClient
with nil to force cleanup.
c.httpClient = nil
if err := c.operationClient.Close(); err != nil {
 return err
}
return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client
 and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 })

```

```

httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
if err != nil {
 return err
}

if err := unmarshal(buf,
resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders,

```



```

hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.License{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource
// exists. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace
// images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy", req.GetProject(),
req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {

```

```

params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended

```

```

for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx,
 hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }
 })
}

```

```

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

```

```

return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the

list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

func (c \*licensesRESTClient) List(ctx context.Context, req \*computepb.ListLicensesRequest, opts

...gax.CallOption) \*LicenseIterator {

```

it := &LicenseIterator{}

```

```

req = proto.Clone(req).(*computepb.ListLicensesRequest)

```

```

unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}

```

```

it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {

```

```

 resp := &computepb.LicensesListResponse{}

```

```

 if pageToken != "" {

```

```

 req.PageToken = proto.String(pageToken)
 }

```

```

}

```

```

 if pageSize > math.MaxInt32

```

```

 {

```

```

 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))

```

```

 } else if pageSize != 0 {

```

```

 req.MaxResults = proto.Uint32(uint32(pageSize))
 }

```

```

}

```

```

baseUrl, err := url.Parse(c.endpoint)

```

```

if err != nil {

```

```

 return nil, "", err
}

```

```

}

```

```

baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

```

```

params := url.Values{}

```

```

if req != nil && req.Filter != nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from
client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

```

```

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken,
 nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds
 := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project", url.QueryEscape(req.GetProject()),
"resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}

```

```

resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if
err := unmarshal(buf, resp); err != nil {
 return err
}

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

```

```

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params",
fmt.Sprintf("%s=%v&%s=%v", "project", url.QueryEscape(req.GetProject()), "resource",
url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err !=
nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//

```



```

// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)

```

```

// Verify that runtime/protoimpl is sufficiently up-to-date.
_ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GranteeType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GranteeType = 3
)

// Enum value maps for Permission_GranteeType.
var (
 Permission_GranteeType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GranteeType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GranteeType) Enum() *Permission_GranteeType {
 p := new(Permission_GranteeType)
 *p = x
 return p
}

func (x Permission_GranteeType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

func (Permission_GranteeType) Descriptor() protorelect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

```

```

}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
 []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)

```

```

*p = x
return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:

```

```

//
// tunedModels/{tuned_model}/permissions/{permission}
// corpora/{corpus}/permissions/{permission}
//
// Output only.
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Optional. Immutable. The type of the grantee.
GranteeType *Permission_GranteeType
`protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission_GranteeType,oneof"
json:"grantee_type,omitempty"`
// Optional. Immutable. The email address of the user of group which this
// permission refers. Field is not set when permission's grantee type is
// EVERYONE.
EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission_Role,oneof"
json:"role,omitempty"`
}

func (x *Permission) Reset() {
*x = Permission{}
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

```

```
// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
 0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,
 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,
 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,
 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,

```

0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,  
0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,  
0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,  
0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,  
0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,  
0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,  
0x01, 0x01, 0x12,  
0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,  
0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,  
0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,  
0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,  
0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,  
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,  
0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,  
0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,  
0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,  
0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,  
0x54,  
0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,  
0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,  
0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,  
0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,  
0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,  
0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,  
0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,  
0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,  
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,  
0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,  
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69,  
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,  
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,  
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,  
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,  
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,  
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,  
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,  
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,  
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,  
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f,  
0x72, 0x6f, 0x6c, 0x65, 0x42,  
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,  
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,  
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,  
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,

```

0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GrantType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}
var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1beta.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
 google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }
func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
 if File_google_ai_generativelanguage_v1beta_permission_proto != nil {

```



```

return
}
file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1beta_permission_proto = out.File
file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
= nil
file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/compute_pb"
)

```

```

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request
 }
 struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.

```

```

// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicenseCodesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseCodeRequest.
}
resp,
err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
)

```

```

func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
 }
 for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3
// source: google/ai/generativelanguage/v1alpha/permission.proto

```

```

package generativelanguagepb

import (
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GrantType int32

const (
 // The default value. This value is unused.
 Permission_GRANT_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANT_TYPE_UNSPECIFIED",
 1: "USER",
 2: "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANT_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 }
)

```

```

 "EVERYONE": 3,
}
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (

```

```

Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
}
Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
}
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string
{
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protorelect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protorelect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protorelect.EnumNumber {
 return protorelect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service

```

```

accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user of group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.
 EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
 json:"email_address,omitempty"`
 // Required. The role granted by this permission.
 Role *Permission_Role
 `protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
 json:"role,omitempty"`
}

func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```



```

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

```

var

File\_google\_ai\_generativelanguage\_v1alpha\_permission\_proto protoreflect.FileDescriptor

```
var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,
 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,
 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,
 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,
 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,
 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,
 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,
 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,
 0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,
 0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,
 0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,
 0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,
 0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,
 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,
 0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,
 0x4e,
 0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,
 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,
 0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,
 0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,
 0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,
 0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,
 0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,
 0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,
 0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,
 0x2e, 0x63, 0x6f, 0x6d,
```

```

0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,
0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,
0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,
0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

```

```

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes
= []any{
 (Permission_GrantType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
}

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.GranteeType
 1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}

func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
 if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
 return
 }
 file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
 }.Build()
 File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
 = nil
 file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
 file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}

// Copyright 2023 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0

```

```

//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.3
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 context "context"
 reflect "reflect"
 sync "sync"

 _ "google.golang.org/genproto/googleapis/api/annotations"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)

```

```

// Verify that runtime/protoimpl is sufficiently up-to-date.
_ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Assignment protocol for a license pool.
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The type of assignment protocol.
 //
 // Types that are assignable to AssignmentType:
 //
 // *AssignmentProtocol_ManualAssignmentType_
 // *AssignmentProtocol_AutoAssignmentType_
 AssignmentType
 isAssignmentProtocol_AssignmentType `protobuf_oneof:"assignment_type"`
}

func (x *AssignmentProtocol) Reset() {
 *x = AssignmentProtocol{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol) ProtoMessage() {}

func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.

```

```

func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType {
 if m != nil {
 return m.AssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
 return x.ManualAssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
 return x.AutoAssignmentType
 }
 return nil
}

type isAssignmentProtocol_AssignmentType interface {
 isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_ struct {
 // Allow manual assignments triggered by administrative operations only.
 ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
 `protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type
AssignmentProtocol_AutoAssignmentType_ struct {
 // Allow automatic assignments triggered by data plane operations.
 AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
 `protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
}

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

```

```

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Identifier. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Required. Assignment protocol for the license pool.
 LicenseAssignmentProtocol *AssignmentProtocol
 `protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
 json:"license_assignment_protocol,omitempty"`
 // Output only. Licenses count that are available to be assigned.
 AvailableLicenseCount int32
 `protobuf:"varint,3,opt,name=available_license_count,json=availableLicenseCount,proto3"
 json:"available_license_count,omitempty"`
 // Output only. Total number of licenses in the pool.
 TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total_license_count,json=totalLicenseCount,proto3"
 json:"total_license_count,omitempty"`
}

func (x *LicensePool) Reset() {
 *x = LicensePool{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensePool) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensePool) ProtoMessage() {}

func (x *LicensePool) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```



```

}

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.
func (*LicensePool) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{1}
}

func (x *LicensePool) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a
// license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

```

```

func (x *GetLicensePoolRequest) Reset() {
 *x = GetLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if
 ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

```

```

// Required. The license pool to update.
//
// The license pool's name field is used to identify the license pool to
// update. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`.
LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
json:"license_pool,omitempty"`
//
Required. The list of fields to update.
UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
json:"update_mask,omitempty"`
}

func (x *UpdateLicensePoolRequest) Reset() {
*x = UpdateLicensePoolRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UpdateLicensePoolRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*UpdateLicensePoolRequest) ProtoMessage() {}

func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor
instead.
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{3}
}

```

```

func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {
 if x != nil {
 return x.LicensePool
 }
 return nil
}

func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

// Request message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.prourement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func
(x *AssignRequest) Reset() {
 *x = AssignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
 mi :=

```

```

&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{4}
}

func (x *AssignRequest) GetParent()
string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

```

```

ms.StoreMessageInfo(mi)
}

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x
 != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{5}
}

// Request message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset()

```

```

{
*x = UnassignRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{6}
}

func (x *UnassignRequest) GetParent() string {
if x != nil {
return
x.Parent
}
return ""
}

func (x *UnassignRequest) GetUnames() []string {
if x != nil {
return x.Unames
}
return nil
}

```

```

// Response message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//

```



```

[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.prourement.v1.License
ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. The maximum number of users to return. The service may return
 // fewer than this value.
 PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3"
 json:"page_size,omitempty"`
 // Optional. A page token, received from a previous `EnumerateLicensedUsers`
 // call. Provide this to retrieve the subsequent page.
 PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

func (x *EnumerateLicensedUsersRequest) Reset() {
 *x = EnumerateLicensedUsersRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[8]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[8]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo()
 == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {

```

```

return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{8}
}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
if x != nil {
return x.Parent
}
return ""
}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
if x != nil {
return x.PageSize
}
return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
if x != nil {
return x.PageToken
}
return ""
}

// A licensed user.
type LicensedUser struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Username.
// Format: `name@domain.com`.
Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
//
// Output only. Timestamp when the license was assigned.
AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3"
json:"assign_time,omitempty"`
// Output only. Timestamp when the license was recently used. This may not be
// the most recent usage time, and will be updated regularly (within 24
// hours).
RecentUsageTime *timestamppb.Timestamp
`protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3"
json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {

```

```

*x = LicensedUser{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if
x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

```

```

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }
 return nil
}

// Response message for
//
// [LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.prourement.v1.License
// ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The list of licensed users.
 LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
 json:"licensed_users,omitempty"`
 // A token that can be sent as `page_token` to retrieve the next page.
 // If this field is omitted, there are no subsequent pages.
 NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
 json:"next_page_token,omitempty"`
}

func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersResponse) ProtoMessage() {}

func
(x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_prourement_v1_license_management_service_proto_msgTypes[10]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)

```

```

 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{10}
}

func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
 return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

```

```

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated:
// Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Optional. The time to live for an inactive license. After this time has
 // passed, the license will be automatically unassigned from the user. Must
 // be at least 7 days, if set. If unset, the license will never expire.
 InactiveLicenseTtl *durationpb.Duration
 `protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
 json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration {
 if x != nil {
 return x.InactiveLicenseTtl
 }
 return nil
}

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protoreflect.FileDescriptor

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{
 0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
 0x6f,
 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
 0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
 0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,

```

0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,  
0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,  
0x62, 0x75, 0x66, 0x2f, 0x64,  
0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,  
0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,  
0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73,  
0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,  
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,  
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01,  
0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42, 0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,



0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,

0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,  
0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69,  
0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,  
0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc5, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,

0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x8b, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x71, 0xda, 0x41, 0x18, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x50, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x40, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,  
0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x2f, 0x2a, 0x7d, 0x12, 0xdf, 0x01,  
0x0a, 0x06, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63,  
0x65,  
0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72,  
0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73,  
0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f,  
0x3a, 0x01, 0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73,  
0x2f, 0x2a, 0x2f, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12,  
0xe7, 0x01, 0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67,

0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x5a, 0xda,  
0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67,  
0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72,  
0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d,  
0x3a, 0x75, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e,  
0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55,  
0x73, 0x65, 0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,  
0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e,  
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73,  
0x65, 0x22, 0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93,  
0x02, 0x4c, 0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d,  
0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f,  
0x2a, 0x2f, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74,  
0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63,  
0xca, 0x41, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63,  
0x6f, 0x6d, 0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f,  
0x61, 0x75, 0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66,  
0x6f, 0x72, 0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,  
0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63,  
0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75,  
0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76,  
0x69, 0x63, 0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f,  
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70,  
0x69, 0x76, 0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70,  
0x62, 0x3b, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa,

```

0x02, 0x2d, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75,
0x64, 0x2e, 0x43,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
0x2e, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca,
0x02, 0x2d, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
0x5c, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea,
0x02, 0x32, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a,
0x3a, 0x43, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75,
0x6d, 0x65, 0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74,
0x3a, 0x3a, 0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
= file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f
unc() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
 })
 return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
 (*AssignmentProtocol)(nil),
 // 0: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 (*LicensePool)(nil),
 // 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
 (*GetLicensePoolRequest)(nil),
 // 2:
 google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 (*UpdateLicensePoolRequest)(nil),
 // 3:
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 (*AssignRequest)(nil),
 // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
 (*AssignResponse)(nil),
 // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
 (*UnassignRequest)(nil),
 // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 (*UnassignResponse)(nil),
 // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 (*EnumerateLicensedUsersRequest)(nil),
 // 8:

```

```

google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
(*LicensedUser)(nil),
 // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
(*EnumerateLicensedUsersResponse)(nil), // 10:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
(*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
(*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
(*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask
(*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
(*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =
[]int32{
 11, // 0:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name
-> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12, // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name -
> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
 13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
google.protobuf.FieldMask
 14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
google.protobuf.Timestamp
 14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name
-> google.protobuf.Timestamp
 9, // 7:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensedUser
 15, // 8:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
ype_name -> google.protobuf.Duration
 2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
> google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 3, // 10:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
google.cloud.commerce.consumer.procurement.v1.AssignRequest
 6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type
-> google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 8, // 13:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_typ
e -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest

```

```

1, // 14:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
1, // 15:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
google.cloud.commerce.consumer.procurement.v1.AssignResponse
7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
google.cloud.commerce.consumer.procurement.v1.UnassignResponse
10, // 18:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_ty
pe
-> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14, // [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}
func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File:
 protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:

```

```

file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
}.Build()
File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs
= nil
}

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// LicenseManagementServiceClient is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {
 // Gets the license pool.
 GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool,
error)
 // Assigns a license to a user.
 Assign(ctx
context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption)
(*EnumerateLicensedUsersResponse, error)
}

type licenseManagementServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
 return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts

```



```

...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool", in, out, opts...)
 if err != nil {
 return nil, err
 }
 return
 out, nil
}

func (c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in
*UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption)
(*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign", in,
out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)

```

```

err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/EnumerateLicensedUsers", in, out,
opts...)
if err != nil {
return nil, err
}
return out, nil
}

// LicenseManagementServiceServer is the server API for LicenseManagementService service.
type LicenseManagementServiceServer interface {
// Gets the license pool.
GetLicensePool(context.Context, *GetLicensePoolRequest) (*LicensePool, error)
// Updates the license pool if one exists for this Order.
UpdateLicensePool(context.Context, *UpdateLicensePoolRequest) (*LicensePool, error)
// Assigns a license to a user.
Assign(context.Context, *AssignRequest)
(*AssignResponse, error)
// Unassigns a license from a user.
Unassign(context.Context, *UnassignRequest) (*UnassignResponse, error)
// Enumerates all users assigned a license.
EnumerateLicensedUsers(context.Context, *EnumerateLicensedUsersRequest)
(*EnumerateLicensedUsersResponse, error)
}

// UnimplementedLicenseManagementServiceServer can be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer struct {
}

func (*UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)
(*UnassignResponse, error) {
return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest) (*EnumerateLicensedUsersResponse, error) {

```

```

return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

func RegisterLicenseManagementServiceServer(s *grpc.Server, srv LicenseManagementServiceServer) {
s.RegisterService(&_LicenseManagementService_serviceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
in := new(GetLicensePoolRequest)
if err := dec(in);
err != nil {
return nil, err
}
if interceptor == nil {
return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, in)
}
info := &grpc.UnaryServerInfo{
Server: srv,
FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/GetLicensePool",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
in := new(UpdateLicensePoolRequest)
if err := dec(in); err != nil {
return nil, err
}
if interceptor == nil {
return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
}
info := &grpc.UnaryServerInfo{
Server: srv,
FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/UpdateLicensePool",
}
handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{}))

```

```

error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(AssignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Assign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv
interface{}, ctx context.Context, dec func(interface{})) error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
 in := new(UnassignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{})) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(EnumerateLicensedUsersRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if
 interceptor == nil {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
 }
}

```

```

info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod:
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
}
return interceptor(ctx, in, info, handler)
}

var _LicenseManagementService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.commerce.consumer.procurement.v1.LicenseManagementService",
 HandlerType: (*LicenseManagementServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "GetLicensePool",
 Handler: _LicenseManagementService_GetLicensePool_Handler,
 },
 {
 MethodName: "UpdateLicensePool",
 Handler: _LicenseManagementService_UpdateLicensePool_Handler,
 },
 {
 MethodName: "Assign",
 Handler: _LicenseManagementService_Assign_Handler,
 },
 {
 MethodName: "Unassign",
 Handler: _LicenseManagementService_Unassign_Handler,
 },
 {
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or



agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.98 x-oauth2 0.30.0

### 1.98.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership  
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.99 gax-go 2.14.2

## 1.99.1 Available under license :

Copyright 2016, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS

BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.100 go-compute-metadata 0.7.0

### 1.100.1 Available under license :

```
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

// [START discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]

package main

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
```

```

"google.golang.org/api/iterator"
)

func main() {
 ctx := context.Background()
 // This snippet
 has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 it := c.ListUserLicenses(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove
 this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
 }
}

// [END discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at

```



```
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"

 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport
 "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newLicenseManagementClientHook clientHook

// LicenseManagementCallOptions contains the retry settings for each method of LicenseManagementClient.
type LicenseManagementCallOptions struct {
 GetLicensePool []gax.CallOption
 UpdateLicensePool []gax.CallOption
 Assign []gax.CallOption
 Unassign []gax.CallOption
 EnumerateLicensedUsers []gax.CallOption
 GetOperation []gax.CallOption
}
```

```

func defaultLicenseManagementGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("cloudcommerceconsumerprocurement.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN:443")
 },
 internaloption.WithDefaultMTLSEndpoint("cloudcommerceconsumerprocurement.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
 }
}

```

```

func
defaultLicenseManagementCallOptions() *LicenseManagementCallOptions {
 return &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
 }
}

```

```

func defaultLicenseManagementRESTCallOptions() *LicenseManagementCallOptions {
 return
 &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
 }
}

```

// internalLicenseManagementClient is an interface that defines the methods available from Cloud Commerce Consumer Procurement API.

```

type internalLicenseManagementClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 GetLicensePool(context.Context, *procurementpb.GetLicensePoolRequest, ...gax.CallOption)
 (*procurementpb.LicensePool, error)
}

```

```

UpdateLicensePool(context.Context, *procurementpb.UpdateLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
Assign(context.Context, *procurementpb.AssignRequest, ...gax.CallOption) (*procurementpb.AssignResponse,
error)
Unassign(context.Context, *procurementpb.UnassignRequest,
...gax.CallOption) (*procurementpb.UnassignResponse, error)
EnumerateLicensedUsers(context.Context, *procurementpb.EnumerateLicensedUsersRequest, ...gax.CallOption)
*LicensedUserIterator
GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
}

```

// LicenseManagementClient is a client for interacting with Cloud Commerce Consumer Procurement API.  
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with  
method calls.

//

// Service for managing licenses.

```

type LicenseManagementClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseManagementClient

```

// The call options for this service.

```

CallOptions *LicenseManagementCallOptions
}

```

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when

// the client is no longer required.

```

func (c *LicenseManagementClient)
Close() error {
 return c.internalClient.Close()
}

```

// setGoogleClientInfo sets the name and version of the application in

// the `x-goog-api-client` header passed on each request. Intended for

// use by Google-written clients.

```

func (c *LicenseManagementClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

```

// Connection returns a connection to the API service.

//

// Deprecated: Connections are now pooled so this method does not always

// return the same resource.

```

func (c *LicenseManagementClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

```

```

// GetLicensePool gets the license pool.
func (c *LicenseManagementClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 return c.internalClient.GetLicensePool(ctx, req, opts...)
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *LicenseManagementClient)
 UpdateLicensePool(ctx context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 return c.internalClient.UpdateLicensePool(ctx, req, opts...)
}

// Assign assigns a license to a user.
func (c *LicenseManagementClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 return c.internalClient.Assign(ctx, req, opts...)
}

// Unassign unassigns a license from a user.
func (c *LicenseManagementClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest, opts
...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 return c.internalClient.Unassign(ctx, req, opts...)
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *LicenseManagementClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 return c.internalClient.EnumerateLicensedUsers(ctx,
 req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *LicenseManagementClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// licenseManagementGRPCClient is a client for interacting with Cloud Commerce Consumer Procurement API
over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type licenseManagementGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

```

```

// Points back to the CallOptions field of the containing LicenseManagementClient
CallOptions **LicenseManagementCallOptions

// The gRPC API client.
licenseManagementClient procurementpb.LicenseManagementServiceClient

operationsClient longrunningpb.OperationsClient

//
The x-goog-* metadata to be sent with each request.
xGoogHeaders []string

logger *slog.Logger
}

// NewLicenseManagementClient creates a new license management service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing licenses.
func NewLicenseManagementClient(ctx context.Context, opts ...option.ClientOption) (*LicenseManagementClient,
error) {
 clientOpts := defaultLicenseManagementGRPCClientOptions()
 if newLicenseManagementClientHook != nil {
 hookOpts, err := newLicenseManagementClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }
 client := LicenseManagementClient{CallOptions: defaultLicenseManagementCallOptions()}

 c := &licenseManagementGRPCClient{
 connPool:
 connPool,
 licenseManagementClient: procurementpb.NewLicenseManagementServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
 }
 c.setGoogleClientInfo()

 client.internalClient = c

```

```

 return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *licenseManagementGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gax", getVersionClient(), "gax", gax.Version, "grpc",
 grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 logger *slog.Logger
}

// NewLicenseManagementRESTClient creates a new license management service rest client.

```

```
//
// Service for managing licenses.
func NewLicenseManagementRESTClient(ctx context.Context,
 opts ...option.ClientOption) (*LicenseManagementClient, error) {
 clientOpts := append(defaultLicenseManagementRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseManagementRESTCallOptions()
 c := &licenseManagementRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseManagementClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseManagementRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://cloudcommerceconsumerprocurement.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://cloudcommerceconsumerprocurement.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
```

```

// the client is no longer required.
func (c *licenseManagementRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient
 = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseManagementRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *licenseManagementGRPCClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.GetLicensePool, req, settings.GRPC, c.logger,
"GetLicensePool")
 return err
 },
 opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 var resp *procurementpb.LicensePool

```



```

err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.UpdateLicensePool, req, settings.GRPC, c.logger,
"UpdateLicensePool")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

func (c *licenseManagementGRPCClient)
Assign(ctx context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption)
(*procurementpb.AssignResponse, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 var resp *procurementpb.AssignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Assign, req, settings.GRPC, c.logger, "Assign")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 hds := []string{"x-goog-request-params",
 fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
 var resp *procurementpb.UnassignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Unassign, req, settings.GRPC, c.logger, "Unassign")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }

```

```

}
return resp, nil
}

func (c *licenseManagementGRPCClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts
=
append((*c.CallOptions).EnumerateLicensedUsers[0:len((*c.CallOptions).EnumerateLicensedUsers):len((*c.CallOptions).EnumerateLicensedUsers)], opts...)
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
resp := &procurementpb.EnumerateLicensedUsersResponse{}
if pageToken != "" {
req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
req.PageSize = math.MaxInt32
} else if pageSize != 0 {
req.PageSize = int32(pageSize)
}
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.licenseManagementClient.EnumerateLicensedUsers, req, settings.GRPC, c.logger,
"EnumerateLicensedUsers")
return err
}, opts...)
if err != nil {
return nil, "", err
}

it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(),
nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

```

```

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *licenseManagementGRPCClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

// GetLicensePool gets the license pool.
func (c *licenseManagementRESTClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)

```

```

hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx,
hds...)
opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicensePool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *licenseManagementRESTClient) UpdateLicensePool(ctx
context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetLicensePool()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)

```

```

if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetLicensePool().GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 params.Add("updateMask", string(field[1:len(field)-1]))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).UpdateLicensePool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Assign assigns a license to a user.
func
(c *licenseManagementRESTClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:assign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial:
true, DiscardUnknown: true}
 resp := &procurementpb.AssignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err

```

```

 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Assign")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Unassign unassigns a license from a user.
func (c *licenseManagementRESTClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err :=
 m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:unassign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)

```

```

opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.UnassignResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Unassign")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *licenseManagementRESTClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 it := &LicensedUserIterator{}
 req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if
 pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)

```



```

}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/v1/%v:enumerateLicensedUsers", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
= settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "EnumerateLicensedUsers")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}

```

```

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetOperation is a utility method from
// google.longrunning.Operations.
func (c *licenseManagementRESTClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }

```

```

httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"

```

```

"net/url"
"time"

computepb "cloud.google.com/go/compute/apiv1/computepb"
gax "github.com/googleapis/gax-go/v2"
"google.golang.org/api/iterator"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
"google.golang.org/protobuf/proto"
)

var
newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
Delete []gax.CallOption
Get []gax.CallOption
GetIamPolicy []gax.CallOption
Insert []gax.CallOption
List []gax.CallOption
SetIamPolicy []gax.CallOption
TestIamPermissions []gax.CallOption
Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
return &LicensesCallOptions{
Delete: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier:
1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
GetIamPolicy: []gax.CallOption{

```

```

gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
}

```

```

Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
TestIamPermissions(context.Context,
*computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

```

// LicensesClient is a client for interacting with Google Compute Engine API.

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

//

// The Licenses API.

type LicensesClient struct {

    // The internal transport-dependent client.

    internalClient internalLicensesClient

    // The call options for this service.

    CallOptions \*LicensesCallOptions

}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when

// the client is no longer required.

func (c \*LicensesClient) Close() error {

    return c.internalClient.Close()

}

// setGoogleClientInfo sets the name and version of the application in

// the `x-goog-api-client` header passed

on each request. Intended for

// use by Google-written clients.

func (c \*LicensesClient) setGoogleClientInfo(keyval ...string) {

    c.internalClient.setGoogleClientInfo(keyval...)

}

// Connection returns a connection to the API service.

//

// Deprecated: Connections are now pooled so this method does not always

// return the same resource.

func (c \*LicensesClient) Connection() \*grpc.ClientConn {

    return c.internalClient.Connection()

}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}
```

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) Get(ctx context.Context, req
*computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}
```

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}
```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}
```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}
```

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}
```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
```

```

error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type
licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicensesClient
 CallOptions **LicensesCallOptions

 logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()
 c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 }

```



```

logger:
 internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {
 return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo sets the name and version of the application in
//
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapis", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {

```

```

 return err
}
return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for use
// only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 })
}

```

```

httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}

```

```

e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
{
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy",
req.GetProject(), req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.

```

```

hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if
settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {

```

```

return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if
err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

params := url.Values{}
if req != nil && req.RequestId != nil {
params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &compute.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err
:= http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil

```

```

 }, opts...)
 if e != nil {
 return nil, e
 }
 op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
 }
 return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for

use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
```

```

...gax.CallOption) *LicenseIterator {
 it := &LicenseIterator{}
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
 resp := &computepb.LicensesListResponse{}
 if pageToken != "" {
 req.PageToken = proto.String(pageToken)
 }
 if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
 } else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())
 }
}

```

```

params := url.Values{}
if req != nil && req.Filter != nil {
 params.Add("filter",
 fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}

```

```

if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header
= headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
}

```



```

 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace
images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 }, httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {

```

```

 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) TestIamPermissions(ctx
context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)

```

```

 opts =
 append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

```

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) Update(ctx context.Context,
 req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 marshalOptions := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := marshal(body)
 if err != nil {
 return nil, err
 }

```

```

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }

```

```

}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

params := url.Values{}
if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}
if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &compute.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

```

```

}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return
op, nil
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability



incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

package compute

import (

```

"bytes"
"context"
"fmt"
"log/slog"
"net/http"
"net/url"
"time"

computepb "cloud.google.com/go/compute/apiv1/computepb"
gax "github.com/googleapis/gax-go/v2"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
)

var
newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {

```

```

Close()
error
setGoogleClientInfo(...string)
Connection() *grpc.ClientConn
Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
(*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return c.internalClient.Close()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud

```

Marketplace images.

```
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}
```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}
```

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

```
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string
```

```
 // The http client.
 httpClient *http.Client
```

```
 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string
```

```
 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions
```

```
 logger *slog.Logger
}
```

```
// NewLicenseCodesRESTClient
creates a new license codes rest client.
//
```

```
// The LicenseCodes API.
```

```
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }
}
```

```
callOpts := defaultLicenseCodesRESTCallOptions()
c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
```

```

 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
}
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gopic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
return
nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
return nil
}

```

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())
```

// Build HTTP headers from client and context metadata.

```
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}
```

```
hds
= append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.LicenseCode{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers
```

```
buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
if err != nil {
 return err
}
```

```
if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}
```

```
return nil
}, opts...)
if e != nil {
 return nil, e
}
```

```

return resp, nil
}

// TestIamPermissions returns permissions that a
// caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v/testIamPermissions",
 req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds =
 append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
 tIamPermissions)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")

```

```

 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1beta/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"

```



```

"google.golang.org/protobuf/encoding/protojson"
)

var
 newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
 }
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.

```

```

type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return
 c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

```

```

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions

 logger *slog.Logger
}

// NewLicenseCodesRESTClient
// creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 }
}

```

```

internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
internaloption.WithDefaultScopes(DefaultAuthScopes()),
internaloption.EnableNewAuthLibrary(),
}
}

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
"x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
return
nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",

```

```
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}}
```

```
hds
= append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.LicenseCode{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
```

// TestIamPermissions returns permissions that

a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }
```

```

}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v/testIamPermissions",
req.GetProject(), req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil

```

```

}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1alpha/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the

```

```

// group.
Permission_GROUP Permission_GranteeType = 2
// Represents access to everyone. No extra information is required.
Permission_EVERYONE Permission_GranteeType = 3
)

// Enum value maps for Permission_GranteeType.
var (
 Permission_GranteeType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GranteeType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GranteeType) Enum() *Permission_GranteeType {
 p := new(Permission_GranteeType)
 *p = x
 return p
}

func (x Permission_GranteeType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GranteeType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GranteeType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GranteeType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GranteeType.Descriptor instead.
func (Permission_GranteeType) EnumDescriptor() ([]byte,

```



```

[]int) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

```

```

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service
// accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user or group which this

```

```

// permission refers. Field is not set when permission's grantee type is
// EVERYONE.
EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
json:"role,omitempty"`
}

func (x *Permission) Reset() {
*x = Permission{}
mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
if x != nil {
return x.Name
}
return ""
}

```

```

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1alpha_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,
 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,
 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,
 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,
 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,
 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,
 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,
 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,
 0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,
 0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,

```

```

0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,
0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,
0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,
0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,
0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,
0x4e,
0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,
0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,
0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,
0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,
0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,
0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,
0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,
0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,
0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,
0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,
0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,
0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,
0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once

```

```

file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
}))
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes
= []any{
(Permission_GrantType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GrantType
(Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
(*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
}
var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->
google.ai.generativelanguage.v1alpha.Permission.GrantType
1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
google.ai.generativelanguage.v1alpha.Permission.Role
2, // [2:2] is the sub-list for method output_type
2, // [2:2] is the sub-list for method input_type
2, // [2:2] is the sub-list for extension type_name
2, // [2:2] is the sub-list for extension extendee
0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}
func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
return
}
file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
NumEnums: 2,
NumMessages: 1,
NumExtensions: 0,
NumServices: 0,

```

```

 },
 GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
 }.Build()
 File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
 = nil
 file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
 file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#ListLicensesRequest.
}
for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package discoveryengine_test

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewUserLicenseClient() {
 ctx := context.Background()
 //

```



This snippet has been automatically generated and should be regarded as a code template only.

// It will require modifications to work:

// - It may require correct/in-range values for request initialization.

// - It may require specifying regional endpoints when creating the service client as shown in:

// [https://pkg.go.dev/cloud.google.com/go#hdr-Client\\_Options](https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options)

c, err := discoveryengine.NewUserLicenseClient(ctx)

if err != nil {

    // TODO: Handle error.

}

defer c.Close()

// TODO: Use client.

\_ = c

}

func ExampleNewUserLicenseRESTClient() {

    ctx := context.Background()

    // This snippet has been automatically generated and should be regarded as a code template only.

    // It will require modifications to work:

    // - It may require correct/in-range values for request initialization.

    // - It may require specifying regional endpoints when creating the service client as shown in:

    // [https://pkg.go.dev/cloud.google.com/go#hdr-Client\\_Options](https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options)

    c, err := discoveryengine.NewUserLicenseRESTClient(ctx)

    if

    err != nil {

        // TODO: Handle error.

    }

    defer c.Close()

    // TODO: Use client.

    \_ = c

}

func ExampleUserLicenseClient\_BatchUpdateUserLicenses() {

    ctx := context.Background()

    // This snippet has been automatically generated and should be regarded as a code template only.

    // It will require modifications to work:

    // - It may require correct/in-range values for request initialization.

    // - It may require specifying regional endpoints when creating the service client as shown in:

    // [https://pkg.go.dev/cloud.google.com/go#hdr-Client\\_Options](https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options)

    c, err := discoveryengine.NewUserLicenseClient(ctx)

    if err != nil {

        // TODO: Handle error.

    }

    defer c.Close()

    req := &discoveryenginepb.BatchUpdateUserLicensesRequest{

        // TODO: Fill request struct fields.

```

// See
https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesRequest.
}
op, err := c.BatchUpdateUserLicenses(ctx, req)
if err != nil {
 //
 TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleUserLicenseClient_ListUserLicenses() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 it := c.ListUserLicenses(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

```

```

// If you need to access the underlying RPC response,
// you can do so by casting the `Response` as below.
// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
}
}

func ExampleUserLicenseClient_CancelOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.CancelOperationRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#CancelOperationRequest.
 }
 err = c.CancelOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleUserLicenseClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }

```

```

}
resp,
err := c.GetOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleUserLicenseClient_ListOperations() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
 }
 it := c.ListOperations(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 //
 TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*longrunningpb.ListOperationsResponse)
 }
}

```

```

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license_service.proto

package discoveryenginepb

import (
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 context "context"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 status "google.golang.org/genproto/googleapis/rpc/status"
 grpc
 "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status1 "google.golang.org/grpc/status"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Request message for
// [UserLicenseService.ListUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses].

```

```

type ListUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The parent [UserStore][] resource name,
 // format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. Requested page size. Server may return fewer items than
 // requested. If unspecified, defaults to 10. The maximum value is 50; values
 // above 50 will be coerced to 50.
 //
 // If this field is negative, an INVALID_ARGUMENT error is returned.
 PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3" json:"page_size,omitempty"`
 // Optional. A page token, received from a previous `ListUserLicenses` call.
 // Provide this to retrieve the subsequent page.
 //
 // When paginating, all other parameters provided to `ListUserLicenses`
 // must match the call that provided the page token.
 PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
 // Optional. Filter for the list request.
 //
 // Supported fields:
 //
 // * `license_assignment_state`
 //
 //
 // Examples:
 //
 // * `license_assignment_state = ASSIGNED` to list assigned user licenses.
 // * `license_assignment_state = NO_LICENSE` to list not licensed users.
 // * `license_assignment_state = NO_LICENSE_ATTEMPTED_LOGIN` to list users
 // who attempted login but no license assigned.
 // * `license_assignment_state != NO_LICENSE_ATTEMPTED_LOGIN` to filter
 // out users who attempted login but no license assigned.
 Filter string `protobuf:"bytes,4,opt,name=filter,proto3" json:"filter,omitempty"`
}

func (x *ListUserLicensesRequest) Reset() {
 *x = ListUserLicensesRequest{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *ListUserLicensesRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

}

func (*ListUserLicensesRequest) ProtoMessage() {}

func (x *ListUserLicensesRequest) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use ListUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*ListUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{0}
}

func (x *ListUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *ListUserLicensesRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *ListUserLicensesRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

func (x *ListUserLicensesRequest) GetFilter() string {
 if x != nil {
 return x.Filter
 }
 return ""
}

```

```

}

// Response message for
// [UserService.ListUserLicenses][google.cloud.discoveryengine.v1.UserService.ListUserLicenses].
type
ListUserLicensesResponse struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// All the customer's
// [UserLicense][google.cloud.discoveryengine.v1.UserLicense]s.
UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
json:"user_licenses,omitempty"`
// A token, which can be sent as `page_token` to retrieve the next page. If
// this field is omitted, there are no subsequent pages.
NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
json:"next_page_token,omitempty"`
}

func (x *ListUserLicensesResponse) Reset() {
*x = ListUserLicensesResponse{}
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *ListUserLicensesResponse) String() string {
return protoimpl.X.MessageStringOf(x)
}

func
(*ListUserLicensesResponse) ProtoMessage() {}

func (x *ListUserLicensesResponse) ProtoReflect() protoreflect.Message {
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use ListUserLicensesResponse.ProtoReflect.Descriptor instead.
func (*ListUserLicensesResponse) Descriptor() ([]byte, []int) {

```



```

return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{1}
}

func (x *ListUserLicensesResponse) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *ListUserLicensesResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Request message for
//
// [UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses]
//
// method.
type BatchUpdateUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The source of the input.
 //
 // Types that are assignable to Source:
 //
 // *BatchUpdateUserLicensesRequest_InlineSource_
 Source isBatchUpdateUserLicensesRequest_Source `protobuf_oneof:"source"`
 // Required. The parent [UserStore][] resource name, format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. If true, if user licenses removed associated license config, the
 // user license will be deleted. By default which is false, the user license
 // will be updated to unassigned state.
 DeleteUnassignedUserLicenses bool
}

func (x *BatchUpdateUserLicensesRequest) Reset() {
 *x = BatchUpdateUserLicensesRequest{}
}

```

```

mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest) ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(),
 []int{2}
}

func (m *BatchUpdateUserLicensesRequest) GetSource() isBatchUpdateUserLicensesRequest_Source {
 if m != nil {
 return m.Source
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest) GetInlineSource() *BatchUpdateUserLicensesRequest_InlineSource {
 if x, ok := x.GetSource().(*BatchUpdateUserLicensesRequest_InlineSource); ok {
 return x.InlineSource
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

```

```

}

func (x *BatchUpdateUserLicensesRequest) GetDeleteUnassignedUserLicenses() bool {
 if x != nil {
 return x.DeleteUnassignedUserLicenses
 }
 return false
}

type isBatchUpdateUserLicensesRequest_Source interface {
 isBatchUpdateUserLicensesRequest_Source()
}

type BatchUpdateUserLicensesRequest_InlineSource_struct {
 // The inline source for the input content for document embeddings.
 InlineSource *BatchUpdateUserLicensesRequest_InlineSource
 `protobuf:"bytes,2,opt,name=inline_source,json=inlineSource,proto3,oneof"`
}

func
 (*BatchUpdateUserLicensesRequest_InlineSource_) isBatchUpdateUserLicensesRequest_Source() {}

// Metadata related to the progress of the
//
[UserService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserService.BatchUpdateUserLicenses]
// operation. This will be returned by the google.longrunning.Operation.metadata
// field.
type BatchUpdateUserLicensesMetadata struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Operation create time.
 CreateTime *timestamppb.Timestamp `protobuf:"bytes,1,opt,name=create_time,json=createTime,proto3"
 json:"create_time,omitempty"`
 // Operation last update time. If the operation is done, this is also the
 // finish time.
 UpdateTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=update_time,json=updateTime,proto3"
 json:"update_time,omitempty"`
 // Count of user licenses successfully updated.
 SuccessCount
 int64 `protobuf:"varint,3,opt,name=success_count,json=successCount,proto3" json:"success_count,omitempty"`
 // Count of user licenses that failed to be updated.
 FailureCount int64 `protobuf:"varint,4,opt,name=failure_count,json=failureCount,proto3"
 json:"failure_count,omitempty"`
}

```

```

func (x *BatchUpdateUserLicensesMetadata) Reset() {
 *x = BatchUpdateUserLicensesMetadata{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesMetadata) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesMetadata) ProtoMessage() {}

func (x *BatchUpdateUserLicensesMetadata) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return
 }
 return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesMetadata.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesMetadata) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{3}
}

func (x *BatchUpdateUserLicensesMetadata) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetSuccessCount() int64 {
 if x != nil {
 return x.SuccessCount
 }

```

```

}
return 0
}

func (x *BatchUpdateUserLicensesMetadata) GetFailureCount() int64 {
 if x != nil {
 return x.FailureCount
 }
 return 0
}

// Response message for
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdat
eUserLicenses]
// method.
type
BatchUpdateUserLicensesResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // UserLicenses successfully updated.
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
 // A sample of errors encountered while processing the request.
 ErrorSamples []*status.Status `protobuf:"bytes,2,rep,name=error_samples,json=errorSamples,proto3"
 json:"error_samples,omitempty"`
}

func (x *BatchUpdateUserLicensesResponse) Reset() {
 *x = BatchUpdateUserLicensesResponse{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesResponse) ProtoMessage() {}

func (x *BatchUpdateUserLicensesResponse) ProtoReflect()
 protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

```

```

if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesResponse.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesResponse) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{4}
}

func (x *BatchUpdateUserLicensesResponse) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *BatchUpdateUserLicensesResponse) GetErrorSamples() []*status.Status {
 if x != nil {
 return x.ErrorSamples
 }
 return nil
}

// The inline source for the input config for BatchUpdateUserLicenses
// method.
type BatchUpdateUserLicensesRequest_InlineSource struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. A list of user licenses to update. Each user license
 // must have a valid
 // [UserLicense.user_principal][google.cloud.discoveryengine.v1.UserLicense.user_principal].
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
 // Optional. The list of fields to update.
 UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
 json:"update_mask,omitempty"`
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) Reset() {
 *x = BatchUpdateUserLicensesRequest_InlineSource{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest_InlineSource)
ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest_InlineSource) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest_InlineSource.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest_InlineSource) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{2, 0}
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

var File_google_cloud_discoveryengine_v1_user_license_service_proto
 protoreflect.FileDescriptor

var file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc = []byte{
 0x0a, 0x3a, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,

```

0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x73,  
0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f,  
0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1c, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f, 0x74, 0x61,  
0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,  
0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,  
0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,  
0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,  
0x1a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,  
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,  
0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x6c, 0x6f, 0x6e,  
0x67, 0x72, 0x75, 0x6e, 0x6e, 0x69, 0x6e, 0x67, 0x2f, 0x6f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69,  
0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2f, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64,  
0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x72, 0x70, 0x63, 0x2f, 0x73, 0x74, 0x61, 0x74, 0x75, 0x73, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc6, 0x01, 0x0a, 0x17, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x12, 0x48, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x30, 0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f,  
0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f,  
0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74,  
0x6f, 0x72, 0x65, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70,  
0x61, 0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a,  
0x0a, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65,  
0x6e, 0x12, 0x1b, 0x0a, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x22, 0x95,  
0x01, 0x0a, 0x18, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75,  
0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18,  
0x01, 0x20, 0x03,  
0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e,  
0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x52, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x26,  
0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65,  
0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65, 0x78, 0x74, 0x50, 0x61, 0x67,



0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x22, 0xe0, 0x03, 0x0a, 0x1e, 0x42, 0x61, 0x74, 0x63, 0x68,  
0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x73, 0x0a, 0x0d, 0x69, 0x6e, 0x6c,  
0x69, 0x6e, 0x65, 0x5f, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b,  
0x32, 0x4c,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,  
0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x2e, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x48, 0x00,  
0x52, 0x0c, 0x69, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x48,  
0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x30,  
0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,  
0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70,  
0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x12, 0x4a, 0x0a, 0x1f, 0x64, 0x65, 0x6c, 0x65,  
0x74, 0x65, 0x5f, 0x75, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65, 0x64, 0x5f, 0x75, 0x73,  
0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x08, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x1c, 0x64, 0x65, 0x6c, 0x65, 0x74, 0x65, 0x55, 0x6e,  
0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x73, 0x1a, 0xa8, 0x01, 0x0a, 0x0c, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53,  
0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x56, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x42, 0x03,  
0xe0, 0x41, 0x02, 0x52,  
0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x40, 0x0a,  
0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73, 0x6b, 0x42,  
0x08, 0x0a, 0x06, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x22, 0xe5, 0x01, 0x0a, 0x1f, 0x42, 0x61,  
0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63,  
0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d, 0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x12, 0x3b, 0x0a,  
0x0b, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f,  
0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a,  
0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x3b, 0x0a, 0x0b, 0x75, 0x70,  
0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32,  
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75,  
0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a, 0x75, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x23, 0x0a, 0x0d, 0x73, 0x75, 0x63, 0x63, 0x65,  
0x73, 0x73, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x03, 0x52, 0x0c,  
0x73, 0x75, 0x63, 0x63, 0x65, 0x73, 0x73, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x23, 0x0a, 0x0d,  
0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x04, 0x20,

0x01, 0x28, 0x03, 0x52, 0x0c, 0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x22, 0xad, 0x01, 0x0a, 0x1f,  
0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74,  
0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x73,  
0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x52, 0x0c, 0x75, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x37, 0x0a, 0x0d, 0x65, 0x72, 0x72, 0x6f,  
0x72, 0x5f, 0x73, 0x61, 0x6d, 0x70, 0x6c, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x0b, 0x32,  
0x12, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x72, 0x70, 0x63, 0x2e, 0x53, 0x74, 0x61,  
0x74, 0x75, 0x73, 0x52, 0x0c, 0x65, 0x72, 0x72, 0x6f, 0x72, 0x53,  
0x61, 0x6d, 0x70, 0x6c, 0x65,  
0x73, 0x32, 0x9a, 0x05, 0x0a, 0x12, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd7, 0x01, 0x0a, 0x10, 0x4c, 0x69, 0x73,  
0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x38, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73,  
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e,  
0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x39, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,  
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73,  
0x65, 0x22, 0x4e, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4,  
0x93, 0x02, 0x3f, 0x12, 0x3d, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x3d, 0x70, 0x72, 0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61,  
0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72,  
0x65, 0x73, 0x2f, 0x2a, 0x7d, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x12, 0xd5, 0x02, 0x0a, 0x17, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x3f,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69,  
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31,  
0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a,  
0x1d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x6c, 0x6f, 0x6e, 0x67, 0x72, 0x75, 0x6e,  
0x6e, 0x69, 0x6e, 0x67, 0x2e, 0x4f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x22, 0xd9,  
0x01, 0xca, 0x41, 0x82, 0x01, 0x0a, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,  
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65,  
0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,  
0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69,

```

0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d,
0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4d, 0x3a, 0x01, 0x2a,
0x22, 0x48, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x70, 0x72,
0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61, 0x74, 0x69, 0x6f,
0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65, 0x73, 0x2f,
0x2a, 0x7d, 0x3a, 0x62, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x1a, 0x52, 0xca, 0x41, 0x1e, 0x64,
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0xd2, 0x41, 0x2e,
0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75, 0x74,
0x68, 0x2f,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72, 0x6d, 0x42, 0x8a,
0x02, 0x0a, 0x23, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,
0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x42, 0x17, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,
0x6e, 0x73, 0x65, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50,
0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64,
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62,
0xa2, 0x02, 0x0f,
0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45, 0x52, 0x59, 0x45, 0x4e, 0x47, 0x49,
0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75,
0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e,
0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c,
0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a,
0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc
)

func
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce.Do(func() {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData
}

```

```

var file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 6)
var file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = []any{
(*ListUserLicensesRequest)(nil), // 0: google.cloud.discoveryengine.v1.ListUserLicensesRequest
(*ListUserLicensesResponse)(nil), // 1: google.cloud.discoveryengine.v1.ListUserLicensesResponse
(*BatchUpdateUserLicensesRequest)(nil), // 2:
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
(*BatchUpdateUserLicensesMetadata)(nil),
// 3: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata
(*BatchUpdateUserLicensesResponse)(nil), // 4:
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse
(*BatchUpdateUserLicensesRequest_InlineSource)(nil), // 5:
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
(*UserLicense)(nil), // 6: google.cloud.discoveryengine.v1.UserLicense
(*timestamppb.Timestamp)(nil), // 7: google.protobuf.Timestamp
(*status.Status)(nil), // 8: google.rpc.Status
(*fieldmaskpb.FieldMask)(nil), // 9: google.protobuf.FieldMask
(*longrunningpb.Operation)(nil), // 10: google.longrunning.Operation
}
var file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = []int32{
6, // 0: google.cloud.discoveryengine.v1.ListUserLicensesResponse.user_licenses:type_name
-> google.cloud.discoveryengine.v1.UserLicense
5, // 1: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.inline_source:type_name ->
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
7, // 2: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.create_time:type_name ->
google.protobuf.Timestamp
7, // 3: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.update_time:type_name ->
google.protobuf.Timestamp
6, // 4: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.user_licenses:type_name ->
google.cloud.discoveryengine.v1.UserLicense
8, // 5: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.error_samples:type_name ->
google.rpc.Status
6, // 6: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.user_licenses:type_name -
> google.cloud.discoveryengine.v1.UserLicense
9, // 7: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.update_mask:type_name
-> google.protobuf.FieldMask
0, // 8: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:input_type ->
google.cloud.discoveryengine.v1.ListUserLicensesRequest
2, // 9: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:input_type ->
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
1, // 10: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:output_type ->
google.cloud.discoveryengine.v1.ListUserLicensesResponse
10, // 11: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:output_type ->
google.longrunning.Operation
10, // [10:12] is the sub-list for method output_type
8, // [8:10] is the sub-list for method input_type
8, // [8:8] is the sub-list for extension type_name

```

```

8, // [8:8] is the sub-list for extension extendee
0, // [0:8] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_service_proto_init()
}
func file_google_cloud_discoveryengine_v1_user_license_service_proto_init() {
if File_google_cloud_discoveryengine_v1_user_license_service_proto != nil {
return
}
file_google_cloud_discoveryengine_v1_user_license_service_proto_init()
file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2].OneofWrappers = []any{
(*BatchUpdateUserLicensesRequest_InlineSource_)(nil),
}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc,
NumEnums: 0,
NumMessages: 6,
NumExtensions: 0,
NumServices: 1,
},
GoTypes: file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes,
DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs,
MessageInfos: file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes,
}.Build()
File_google_cloud_discoveryengine_v1_user_license_service_proto
= out.File
file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc = nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = nil
}

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// UserLicenseServiceClient is the client API for UserLicenseService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type UserLicenseServiceClient interface {

```

```

// Lists the User Licenses.
ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest,
opts ...grpc.CallOption) (*ListUserLicensesResponse, error)
// Updates the User License.
// This method is used for batch assign/unassign licenses to users.
BatchUpdateUserLicenses(ctx context.Context, in *BatchUpdateUserLicensesRequest, opts ...grpc.CallOption)
(*longrunningpb.Operation, error)
}

type userLicenseServiceClient struct {
cc grpc.ClientConnInterface
}

func NewUserLicenseServiceClient(cc grpc.ClientConnInterface) UserLicenseServiceClient {
return &userLicenseServiceClient{cc}
}

func (c *userLicenseServiceClient) ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest, opts
...grpc.CallOption) (*ListUserLicensesResponse, error) {
out := new(ListUserLicensesResponse)
err := c.cc.Invoke(ctx, "/google.cloud.discoveryengine.v1.UserLicenseService/ListUserLicenses", in, out, opts...)
if err != nil {
return nil, err
}
return out, nil
}

func (c *userLicenseServiceClient) BatchUpdateUserLicenses(ctx context.Context, in
*BatchUpdateUserLicensesRequest,
opts ...grpc.CallOption) (*longrunningpb.Operation, error) {
out := new(longrunningpb.Operation)
err := c.cc.Invoke(ctx, "/google.cloud.discoveryengine.v1.UserLicenseService/BatchUpdateUserLicenses", in, out,
opts...)
if err != nil {
return nil, err
}
return out, nil
}

// UserLicenseServiceServer is the server API for UserLicenseService service.
type UserLicenseServiceServer interface {
// Lists the User Licenses.
ListUserLicenses(context.Context, *ListUserLicensesRequest) (*ListUserLicensesResponse, error)
// Updates the User License.
// This method is used for batch assign/unassign licenses to users.
BatchUpdateUserLicenses(context.Context, *BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error)
}

```

```

// UnimplementedUserLicenseServiceServer can be embedded to have forward compatible implementations.
type UnimplementedUserLicenseServiceServer struct {
}

func (*UnimplementedUserLicenseServiceServer) ListUserLicenses(context.Context, *ListUserLicensesRequest)
(*ListUserLicensesResponse, error) {
 return nil, status1.Errorf(codes.Unimplemented, "method ListUserLicenses not implemented")
}

func (*UnimplementedUserLicenseServiceServer) BatchUpdateUserLicenses(context.Context,
*BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error) {
 return nil, status1.Errorf(codes.Unimplemented, "method BatchUpdateUserLicenses not implemented")
}

func RegisterUserLicenseServiceServer(s *grpc.Server, srv UserLicenseServiceServer) {
 s.RegisterService(&_UserLicenseService_serviceDesc, srv)
}

func _UserLicenseService_ListUserLicenses_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(ListUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.discoveryengine.v1.UserLicenseService/ListUserLicenses",
 }
 handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, req.(*ListUserLicensesRequest))
}
 return interceptor(ctx, in, info, handler)
}

func _UserLicenseService_BatchUpdateUserLicenses_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(BatchUpdateUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,

```

```

FullMethod: "/google.cloud.discoveryengine.v1.UserLicenseService/BatchUpdateUserLicenses",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, req.(*BatchUpdateUserLicensesRequest))
}
return interceptor(ctx,
in, info, handler)
}

var _UserLicenseService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.discoveryengine.v1.UserLicenseService",
 HandlerType: (*UserLicenseServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "ListUserLicenses",
 Handler: _UserLicenseService_ListUserLicenses_Handler,
 },
 {
 MethodName: "BatchUpdateUserLicenses",
 Handler: _UserLicenseService_BatchUpdateUserLicenses_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/discoveryengine/v1/user_license_service.proto",
}

// Copyright 2023 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software

```



```

// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{

```

```

// TODO:
Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseCodeRequest.
}
resp, err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseCodeRequest.
 }
 resp,
 err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}
// Copyright 2024 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,

```

```

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
 }

```

```

for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license.proto

package discoveryenginepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect
 "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// License assignment state enumeration.

```

```

type UserLicense_LicenseAssignmentState int32

const (
 // Default value.
 UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED UserLicense_LicenseAssignmentState = 0
 // License assigned to the user.
 UserLicense_ASSIGNED UserLicense_LicenseAssignmentState = 1
 // No license assigned to the user.
 // Deprecated, translated to NO_LICENSE.
 UserLicense_UNASSIGNED UserLicense_LicenseAssignmentState = 2
 // No license assigned to the user.
 UserLicense_NO_LICENSE UserLicense_LicenseAssignmentState = 3
 // User attempted to login but no license assigned to the user.
 // This state is only used for no user first time login attempt
 // but cannot
 // get license assigned.
 // Users already logged in but cannot get license assigned will be assigned
 // NO_LICENSE state(License could be unassigned by admin).
 UserLicense_NO_LICENSE_ATTEMPTED_LOGIN UserLicense_LicenseAssignmentState = 4
)

// Enum value maps for UserLicense_LicenseAssignmentState.
var (
 UserLicense_LicenseAssignmentState_name = map[int32]string{
 0: "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED",
 1: "ASSIGNED",
 2: "UNASSIGNED",
 3: "NO_LICENSE",
 4: "NO_LICENSE_ATTEMPTED_LOGIN",
 }
 UserLicense_LicenseAssignmentState_value = map[string]int32{
 "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED": 0,
 "ASSIGNED": 1,
 "UNASSIGNED": 2,
 "NO_LICENSE": 3,
 "NO_LICENSE_ATTEMPTED_LOGIN": 4,
 }
)

func (x UserLicense_LicenseAssignmentState) Enum() *UserLicense_LicenseAssignmentState {
 p := new(UserLicense_LicenseAssignmentState)
 *p = x
 return p
}

func (x UserLicense_LicenseAssignmentState)
String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

```

```

}

func (UserLicense_LicenseAssignmentState) Descriptor() protoreflect.EnumDescriptor {
 return file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0].Descriptor()
}

func (UserLicense_LicenseAssignmentState) Type() protoreflect.EnumType {
 return &file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0]
}

func (x UserLicense_LicenseAssignmentState) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use UserLicense_LicenseAssignmentState.Descriptor instead.
func (UserLicense_LicenseAssignmentState) EnumDescriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0, 0}
}

// User License information assigned by the admin.
type UserLicense struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 //
 // Required. Immutable. The user principal of the User, could be email address
 // or other principal identifier. This field is immutable. Admin assign
 // licenses based on the user principal.
 UserPrincipal string `protobuf:"bytes,1,opt,name=user_principal,json=userPrincipal,proto3"
 json:"user_principal,omitempty"`
 // Optional. The user profile.
 // We use user full name(First name + Last name) as user profile.
 UserProfile string `protobuf:"bytes,3,opt,name=user_profile,json=userProfile,proto3"
 json:"user_profile,omitempty"`
 // Output only. License assignment state of the user.
 // If the user is assigned with a license config, the user login will be
 // assigned with the license;
 // If the user's license assignment state is unassigned or unspecified, no
 // license config will be associated to the user;
 LicenseAssignmentState UserLicense_LicenseAssignmentState
 `protobuf:"varint,4,opt,name=license_assignment_state,json=licenseAssignmentState,proto3,enum=google.cloud.discoveryengine.v1.UserLicense_LicenseAssignmentState"
 json:"license_assignment_state,omitempty"`
 // Optional. The full resource name of the Subscription(LicenseConfig)
 // assigned to the user.
 LicenseConfig string `protobuf:"bytes,5,opt,name=license_config,json=licenseConfig,proto3"
 json:"license_config,omitempty"`
}

```

```

// Output only. User created timestamp.
CreateTime *timestamppb.Timestamp `protobuf:"bytes,6,opt,name=create_time,json=createTime,proto3"
json:"create_time,omitempty"`
// Output only. User update timestamp.
UpdateTime *timestamppb.Timestamp `protobuf:"bytes,7,opt,name=update_time,json=updateTime,proto3"
json:"update_time,omitempty"`
// Output only. User last logged in time.
// If the user has not logged in yet, this field will be empty.
LastLoginTime *timestamppb.Timestamp
`protobuf:"bytes,8,opt,name=last_login_time,json=lastLoginTime,proto3" json:"last_login_time,omitempty"`
}

func (x *UserLicense) Reset() {
*x = UserLicense{}
mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
ms
:= protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UserLicense) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*UserLicense) ProtoMessage() {}

func (x *UserLicense) ProtoReflect() protoreflect.Message {
mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use UserLicense.ProtoReflect.Descriptor instead.
func (*UserLicense) Descriptor() ([]byte, []int) {
return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0}
}

func (x *UserLicense) GetUserPrincipal() string {
if x != nil {
return x.UserPrincipal
}
return ""
}

```

```

}

func (x *UserLicense) GetUserProfile() string {
 if x != nil {
 return x.UserProfile
 }
 return ""
}

func (x *UserLicense) GetLicenseAssignmentState()
 UserLicense_LicenseAssignmentState {
 if x != nil {
 return x.LicenseAssignmentState
 }
 return UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED
}

func (x *UserLicense) GetLicenseConfig() string {
 if x != nil {
 return x.LicenseConfig
 }
 return ""
}

func (x *UserLicense) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}

func (x *UserLicense) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}

func (x *UserLicense) GetLastLoginTime() *timestamppb.Timestamp {
 if x != nil {
 return x.LastLoginTime
 }
 return nil
}

var File_google_cloud_discoveryengine_v1_user_license_proto protoreflect.FileDescriptor

var file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = []byte{

```



0x0a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,  
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,  
0x31,  
0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f,  
0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,  
0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,  
0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72,  
0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,  
0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,  
0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f,  
0x74, 0x6f, 0x22, 0xa6, 0x05, 0x0a, 0x0b, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x12, 0x2d, 0x0a, 0x0e,  
0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x69, 0x6e, 0x63,  
0x69, 0x70, 0x61, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x05, 0xe0,  
0x41, 0x02, 0x52, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x50, 0x72, 0x69, 0x6e, 0x63, 0x69, 0x70, 0x61,  
0x6c, 0x12, 0x26, 0x0a, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x6f, 0x66, 0x69, 0x6c,  
0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x0b, 0x75, 0x73,  
0x65, 0x72, 0x50, 0x72, 0x6f, 0x66, 0x69, 0x6c, 0x65, 0x12, 0x82, 0x01, 0x0a, 0x18, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74,  
0x5f, 0x73, 0x74, 0x61, 0x74, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x43, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e,  
0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74,  
0x65, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x16, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74, 0x65, 0x12, 0x5b,  
0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x6e, 0x66, 0x69, 0x67,  
0x18, 0x05, 0x20, 0x01, 0x28, 0x09, 0x42, 0x34, 0xe0, 0x41, 0x01, 0xfa, 0x41, 0x2e, 0x0a, 0x2c,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x52, 0x0d, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x12, 0x40, 0x0a, 0x0b, 0x63,  
0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x06, 0x20, 0x01, 0x28, 0x0b,  
0x32,  
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,  
0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41,  
0x03, 0x52, 0x0a, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x07, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x47, 0x0a, 0x0f, 0x6c, 0x61, 0x73, 0x74, 0x5f, 0x6c, 0x6f, 0x67, 0x69, 0x6e, 0x5f, 0x74, 0x69,  
0x6d, 0x65, 0x18, 0x08, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73,  
0x74, 0x61, 0x6d, 0x70, 0x42,

```

0x03, 0xe0, 0x41, 0x03, 0x52, 0x0d, 0x6c, 0x61, 0x73, 0x74, 0x4c,
0x6f, 0x67, 0x69, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x22, 0x90, 0x01, 0x0a, 0x16, 0x4c, 0x69, 0x63,
0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74,
0x61, 0x74, 0x65, 0x12, 0x28, 0x0a, 0x24, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41,
0x53, 0x53, 0x49, 0x47, 0x4e, 0x4d, 0x45, 0x4e, 0x54, 0x5f, 0x53, 0x54, 0x41, 0x54, 0x45, 0x5f,
0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x0c, 0x0a,
0x08, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x01, 0x12, 0x0e, 0x0a, 0x0a, 0x55,
0x4e, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x02, 0x12, 0x0e, 0x0a, 0x0a, 0x4e,
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x10, 0x03, 0x12, 0x1e, 0x0a, 0x1a, 0x4e,
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41, 0x54, 0x54, 0x45, 0x4d, 0x50,
0x54, 0x45, 0x44, 0x5f, 0x4c, 0x4f, 0x47, 0x49, 0x4e, 0x10,
0x04, 0x42, 0x83, 0x02, 0x0a, 0x23,
0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,
0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65,
0x2e, 0x76, 0x31, 0x42, 0x10, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69,
0x76, 0x31, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,
0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,
0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0xa2, 0x02, 0x0f, 0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45,
0x52, 0x59, 0x45, 0x4e, 0x47, 0x49, 0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67,
0x6c,
0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f,
0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76,
0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69,
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56,
0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc
)

func file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce.Do(func()
 {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData
}

```

```

var file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes = make([]protoimpl.EnumInfo, 1)

```

```

var file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = []any{
 (UserLicense_LicenseAssignmentState)(0), // 0:
 google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 (*UserLicense)(nil), // 1: google.cloud.discoveryengine.v1.UserLicense
 (*timestamppb.Timestamp)(nil), // 2: google.protobuf.Timestamp
}
var file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = []int32{
 0, // 0: google.cloud.discoveryengine.v1.UserLicense.license_assignment_state:type_name
 -> google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 2, // 1: google.cloud.discoveryengine.v1.UserLicense.create_time:type_name -> google.protobuf.Timestamp
 2, // 2: google.cloud.discoveryengine.v1.UserLicense.update_time:type_name -> google.protobuf.Timestamp
 2, // 3: google.cloud.discoveryengine.v1.UserLicense.last_login_time:type_name -> google.protobuf.Timestamp
 4, // [4:4] is the sub-list for method output_type
 4, // [4:4] is the sub-list for method input_type
 4, // [4:4] is the sub-list for extension type_name
 4, // [4:4] is the sub-list for extension extendee
 0, // [0:4] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_proto_init() }
func file_google_cloud_discoveryengine_v1_user_license_proto_init() {
 if File_google_cloud_discoveryengine_v1_user_license_proto != nil {
 return
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc,
 NumEnums: 1,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_cloud_discoveryengine_v1_user_license_proto_goTypes,
 DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs,
 EnumInfos: file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes,
 MessageInfos: file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes,
 }.Build()
 File_google_cloud_discoveryengine_v1_user_license_proto = out.File
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = nil
 file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = nil
 file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//

```

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
```

```

// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c,
err := procurement.NewLicenseManagementRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleLicenseManagementClient_Assign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
 }
 resp, err := c.Assign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 //
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicensedUsersRequest.
}
it := c.EnumerateLicensedUsers(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 //
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
}
}

func ExampleLicenseManagementClient_GetLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //

```

See

<https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRequest>.

```
}
```

```
resp, err := c.GetLicensePool(ctx, req)
```

```
if err != nil {
```

```
 // TODO: Handle error.
```

```
}
```

```
// TODO: Use resp.
```

```
_ = resp
```

```
}
```

```
func ExampleLicenseManagementClient_Unassign() {
```

```
 ctx := context.Background()
```

```
 // This snippet has been automatically generated and should be regarded as a code template only.
```

```
 // It will require modifications to work:
```

```
 // - It may require correct/in-range values for request initialization.
```

```
 // - It may require specifying regional endpoints when creating the service client as shown in:
```

```
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
```

```
 c, err := procurement.NewLicenseManagementClient(ctx)
```

```
 if err != nil {
```

```
 // TODO: Handle error.
```

```
 }
```

```
 defer c.Close()
```

```
 req := &procurementpb.UnassignRequest{
```

```
 // TODO: Fill request struct fields.
```

```
 // See
```

<https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest>.

```
 }
```

```
 resp,
```

```
 err := c.Unassign(ctx, req)
```

```
 if err != nil {
```

```
 // TODO: Handle error.
```

```
 }
```

```
 // TODO: Use resp.
```

```
 _ = resp
```

```
}
```

```
func ExampleLicenseManagementClient_UpdateLicensePool() {
```

```
 ctx := context.Background()
```

```
 // This snippet has been automatically generated and should be regarded as a code template only.
```

```
 // It will require modifications to work:
```

```
 // - It may require correct/in-range values for request initialization.
```

```
 // - It may require specifying regional endpoints when creating the service client as shown in:
```

```
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
```

```
 c, err := procurement.NewLicenseManagementClient(ctx)
```

```
 if err != nil {
```

```

// TODO: Handle error.
}
defer c.Close()

req := &procurementpb.UpdateLicensePoolRequest{
// TODO: Fill request struct fields.
// See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
}
resp, err := c.UpdateLicensePool(ctx, req)
if err != nil {
// TODO: Handle error.
}
//
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_GetOperation() {
ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
// TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.GetOperationRequest{
// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
}
resp, err := c.GetOperation(ctx, req)
if err != nil {
// TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:



\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

//go:build go1.23

package compute\_test

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

```
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 //
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}
```

```

}
defer c.Close()

req := &computepb.DeleteLicenseRequest{
 //
 TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#DeleteLicenseRequest.
}
op, err := c.Delete(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.

```

```

// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetIamPolicyLicenseRequest.
}
resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

}
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded
 as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#ListLicensesRequest.
 }
 it := c.List(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ =
 = it.Response.(*computepb.LicensesListResponse)
 }
}

```

```

func ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:

```

```

// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#SetIamPolicyLicenseRequest.
}
resp, err := c.SetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.

```

```

// It will require
modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#UpdateLicenseRequest.
}
op, err := c.Update(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/compute"

```

)

```
func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request
 struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
```



```

// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicenseCodesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#TestIamPermissionsLicenseCodeRequest.
}
resp,
err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"

```

)

```
func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
 }
 for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package discoveryengine
```

```

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 "cloud.google.com/go/longrunning"
 lroauto "cloud.google.com/go/longrunning/autogen"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax
 "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/grpc/codes"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newUserLicenseClientHook clientHook

// UserLicenseCallOptions contains the retry settings for each method of UserLicenseClient.
type UserLicenseCallOptions struct {
 ListUserLicenses []gax.CallOption
 BatchUpdateUserLicenses []gax.CallOption
 CancelOperation []gax.CallOption
 GetOperation []gax.CallOption
 ListOperations []gax.CallOption
}

func defaultUserLicenseGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("discoveryengine.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("discoveryengine.UNIVERSE_DOMAIN:443"),
 internaloption.WithDefaultMTLSEndpoint("discoveryengine.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 }
}

```

```

option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
}
}

```

```

func
defaultUserLicenseCallOptions() *UserLicenseCallOptions {
return &UserLicenseCallOptions{
 ListUserLicenses: []gax.CallOption{},
 BatchUpdateUserLicenses: []gax.CallOption{},
 CancelOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier:
1.30,
 })
 }),
 },
 GetOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
 },
 ListOperations: []gax.CallOption{
 gax.WithTimeout(300000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 30000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
 },
}
}

```

```

}

func defaultUserLicenseRESTCallOptions() *UserLicenseCallOptions {
return &UserLicenseCallOptions{
ListUserLicenses: []gax.CallOption{},
BatchUpdateUserLicenses: []gax.CallOption{},
CancelOperation: []gax.CallOption{
gax.WithTimeout(30000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer
{
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 10000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusServiceUnavailable)
}),
},
GetOperation: []gax.CallOption{
gax.WithTimeout(30000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 10000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusServiceUnavailable)
}),
},
ListOperations: []gax.CallOption{
gax.WithTimeout(300000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 30000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusServiceUnavailable)
}),
},
}
}

// internalUserLicenseClient is an interface that defines the methods available from Discovery Engine API.
type internalUserLicenseClient
interface {
Close() error
setGoogleClientInfo(...string)

```

```

Connection() *grpc.ClientConn
ListUserLicenses(context.Context, *discoveryenginepb.ListUserLicensesRequest, ...gax.CallOption)
*UserLicenseIterator
BatchUpdateUserLicenses(context.Context, *discoveryenginepb.BatchUpdateUserLicensesRequest,
...gax.CallOption) (*BatchUpdateUserLicensesOperation, error)
BatchUpdateUserLicensesOperation(name string) *BatchUpdateUserLicensesOperation
CancelOperation(context.Context, *longrunningpb.CancelOperationRequest, ...gax.CallOption) error
GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
ListOperations(context.Context, *longrunningpb.ListOperationsRequest, ...gax.CallOption) *OperationIterator
}

// UserLicenseClient is a client for interacting with Discovery Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// Service for managing
// User Licenses.
type UserLicenseClient struct {
 // The internal transport-dependent client.
 internalClient internalUserLicenseClient

 // The call options for this service.
 CallOptions *UserLicenseCallOptions

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient *lroauto.OperationsClient
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *UserLicenseClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *UserLicenseClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns

```

```

a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *UserLicenseClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// ListUserLicenses lists the User Licenses.
func (c *UserLicenseClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
 return c.internalClient.ListUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicenses updates the User License.
// This method is used for batch assign/unassign licenses to users.
func (c *UserLicenseClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 return c.internalClient.BatchUpdateUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
//
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *UserLicenseClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 return c.internalClient.BatchUpdateUserLicensesOperation(name)
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest,
opts ...gax.CallOption) error {
 return c.internalClient.CancelOperation(ctx, req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts
...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) ListOperations(ctx context.Context,
req *longrunningpb.ListOperationsRequest, opts ...gax.CallOption) *OperationIterator {
 return c.internalClient.ListOperations(ctx, req, opts...)
}

```

```

// userLicenseGRPCClient is a client for interacting with Discovery Engine API over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing UserLicenseClient
 CallOptions **UserLicenseCallOptions

 // The gRPC API client.
 userLicenseClient discoveryenginepb.UserLicenseServiceClient

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 operationsClient longrunningpb.OperationsClient

 // The x-goog-* metadata
 // to be sent with each request.
 xGoogHeaders []string

 logger *slog.Logger
}

// NewUserLicenseClient creates a new user license service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing User Licenses.
func NewUserLicenseClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := defaultUserLicenseGRPCClientOptions()
 if newUserLicenseClientHook != nil {
 hookOpts, err := newUserLicenseClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }
 client := UserLicenseClient{CallOptions: defaultUserLicenseCallOptions()}

```



```

c := &userLicenseGRPCClient{
 connPool: connPool,
 userLicenseClient: discoveryenginepb.NewUserLicenseServiceClient(connPool),
 CallOptions:
 &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
}
c.setGoogleClientInfo()

client.internalClient = c

client.LROClient, err = lroauto.NewOperationsClient(ctx, gtransport.WithConnPool(connPool))
if err != nil {
 // This error "should not happen", since we are just reusing old connection pool
 // and never actually need to dial.
 // If this does happen, we could leak connp. However, we cannot close conn:
 // If the user invoked the constructor with option.WithGRPCConn,
 // we would close a connection that's still in use.
 // TODO: investigate error conditions.
 return nil, err
}
c.LROClient = &client.LROClient
return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *userLicenseGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc", grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when

```

```

// the client is no longer required.
func (c *userLicenseGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // LROClient is used internally
 // to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing UserLicenseClient
 CallOptions **UserLicenseCallOptions

 logger *slog.Logger
}

// NewUserLicenseRESTClient creates a new user license service rest client.
//
// Service for managing User Licenses.
func NewUserLicenseRESTClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := append(defaultUserLicenseRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultUserLicenseRESTCallOptions()
 c := &userLicenseRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

```

```

lroOpts
:= []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opClient, err := lroauto.NewOperationsRESTClient(ctx, lroOpts...)
if err != nil {
 return nil, err
}
c.LROClient = &opClient

return &UserLicenseClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultUserLicenseRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://discoveryengine.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://discoveryengine.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://discoveryengine.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo
// sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapis", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *userLicenseRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return nil
}

// Connection returns a connection to the API service.
//

```

```

// Deprecated: This method always returns nil.
func (c *userLicenseRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *userLicenseGRPCClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest,
opts ...gax.CallOption) *UserLicenseIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).ListUserLicenses[0:len((*c.CallOptions).ListUserLicenses):len((*c.CallOptions).ListUser
Licenses)], opts...)
 it := &UserLicenseIterator{}
 req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
 resp := &discoveryenginepb.ListUserLicensesResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx,
c.userLicenseClient.ListUserLicenses, req, settings.GRPC, c.logger, "ListUserLicenses")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
 }
 fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
 }
}

```

```

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *userLicenseGRPCClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders,
hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).BatchUpdateUserLicenses[0:len((*c.CallOptions).BatchUpdateUserLicenses):len((*c.Call
Options).BatchUpdateUserLicenses)], opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.userLicenseClient.BatchUpdateUserLicenses, req, settings.GRPC, c.logger,
"BatchUpdateUserLicenses")
return err
}, opts...)
if err != nil {
return nil, err
}
}
return &BatchUpdateUserLicensesOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, resp),
}, nil
}

func (c *userLicenseGRPCClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx,
hds...)
opts =
append((*c.CallOptions).CancelOperation[0:len((*c.CallOptions).CancelOperation):len((*c.CallOptions).CancelOp
eration)], opts...)
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
_, err = executeRPC(ctx, c.operationsClient.CancelOperation, req, settings.GRPC, c.logger, "CancelOperation")
return err
}, opts...)

```

```

return err
}

func (c *userLicenseGRPCClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context,
settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

func (c *userLicenseGRPCClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,
opts ...gax.CallOption) *OperationIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).ListOperations[0:len((*c.CallOptions).ListOperations):len((*c.CallOptions).ListOperations)], opts...)
it := &OperationIterator{}
req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
resp := &longrunningpb.ListOperationsResponse{}
if pageToken !=
"" {
req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
req.PageSize = math.MaxInt32
} else if pageSize != 0 {
req.PageSize = int32(pageSize)
}
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

var err error
resp, err = executeRPC(ctx, c.operationsClient.ListOperations, req, settings.GRPC, c.logger, "ListOperations")
return err
}, opts...)
if err != nil {
return nil, "", err
}

it.Response = resp
return resp.GetOperations(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// ListUserLicenses lists the
// User Licenses.
func (c *userLicenseRESTClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
it := &UserLicenseIterator{}
req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
resp := &discoveryenginepb.ListUserLicensesResponse{}
if pageToken != "" {
req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
req.PageSize = math.MaxInt32
} else if pageSize != 0 {
req.PageSize = int32(pageSize)
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, "", err
}

```

```

baseUrl.Path += fmt.Sprintf("/v1/%v/userLicenses", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v",
req.GetFilter()))
}
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListUserLicenses")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response
= resp
return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
}

```



```

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// BatchUpdateUserLicenses updates the User License.
// This method is used for batch assign/unassign licenses to users.
func (c *userLicenseRESTClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:batchUpdateUserLicenses", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

if settings.Path != "" {
 baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err
:= executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "BatchUpdateUserLicenses")
if err != nil {
 return err
}
if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:cancel", req.GetName())

 params := url.Values{}

```

```

params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery
= params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
return gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 _, err = executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "CancelOperation")
 return err
}, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption)
(*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)

```

```

 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET",
baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }

 if err := unmarshalOptions.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,
opts ...gax.CallOption) *OperationIterator {
 it := &OperationIterator{}
 req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{}
 if pageToken != "" {
 req.PageToken
= pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32

```

```

 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/operations", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
= settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListOperations")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {

```

```

 return nil, "", e
}
it.Response = resp
return resp.GetOperations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// BatchUpdateUserLicensesOperation returns a new
// BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *userLicenseGRPCClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 }
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *userLicenseRESTClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.

```

```
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.
```

```
package compute
```

```
import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1beta/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)
```

```
var
```

```
newLicensesClientHook clientHook
```

```
// LicensesCallOptions contains the retry settings for each method of LicensesClient.
```

```
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
 Update []gax.CallOption
}
```

```

}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
return &LicensesCallOptions{
Delete: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier:
1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
GetIamPolicy: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
Insert: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
SetIamPolicy: []gax.CallOption{

```



```

 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
 TestIamPermissions(context.Context,
*computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
 Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.
 CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {

```

```

return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header
// passed on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context,
 req *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available

```

in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}
```

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}
```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}
```

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}
```

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

type

licensesRESTClient struct {

// The http endpoint to connect to.

endpoint string

// The http client.

httpClient \*http.Client

// operationClient is used to call the operation-specific management service.

operationClient \*GlobalOperationsClient

// The x-goog-\* headers to be sent with each request.

xGoogHeaders []string

```

// Points back to the CallOptions field of the containing LicensesClient
CallOptions **LicensesCallOptions

logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()
 c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
 }
 opC, err := NewGlobalOperationsRESTClient(ctx, o...)
 if err != nil {
 return nil, err
 }
 c.operationClient = opC

 return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

```

```

}
}

// setGoogleClientInfo sets the name and version of the application
in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {
 return err
 }
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for
// use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
}

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

```

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
who are creating
Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil

```

```

}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/getIamPolicy",
 req.GetProject(), req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }
 }
 })

```



```

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err
 := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

if settings.Path != "" {
 baseUrl.Path = settings.Path
}
httpReq,
err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended

for use only by third-party partners who are creating Cloud Marketplace images.

func (c \*licensesRESTClient) List(ctx context.Context, req \*compute.LicenseListRequest, opts

...gax.CallOption) \*LicenseIterator {

it := &LicenseIterator{}

req = proto.Clone(req).(\*compute.LicenseListRequest)

unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}

it.InternalFetch = func(pageSize int, pageToken string) ([]\*compute.License, string, error) {

resp := &compute.LicenseListResponse{}

if pageToken != "" {

```

 req.PageToken = proto.String(pageToken)
}
if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
} else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

params := url.Values{}
if req != nil && req.Filter != nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")

```

```

 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating
Cloud Marketplace images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }

```

```

baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
TestIamPermissions(ctx context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts
...gax.CallOption) (*computepb.TestPermissionsResponse, error) {

```

```

m := protojson.MarshalOptions{AllowPartial: true}
body := req.GetTestPermissionsRequestResource()
jsonReq, err := m.Marshal(body)
if err != nil {
 return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil

```

```

 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
 if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds
 = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path

```

```

 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project:
req.GetProject(),
 },
}
return op, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:

```



```
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GrantType int32

const (
 // The default value. This value is unused.
 Permission_GRANT_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANT_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANT_TYPE_UNSPECIFIED": 0,
 }
)
```

```

"USER": 1,
"GROUP": 2,
"EVERYONE": 3,
}
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

```

```

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform

```

```

// specific actions on PaLM API resources. To make them available to users,
// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user or group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.
 EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
 json:"email_address,omitempty"`
 // Required. The role granted by this permission.
 Role *Permission_Role
 `protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission_Role,oneof"
 json:"role,omitempty"`
}

func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {

```

```

return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
}

```

```
return Permission_ROLE_UNSPECIFIED
}
```

```
var
```

```
File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor
```

```
var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
 0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,
 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,
 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,
 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,
 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,
 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,
 0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,
 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,
 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,
 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,
 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12,
 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,
 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,
 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,
 0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,
 0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,
 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,
 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,
 0x54, 0x59,
 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,
 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,
 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,
 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,
 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,
 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,
 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,
 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,
 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
```

```

0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42,
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GranteeType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GranteeType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
}

```

```

(*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}
var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1beta.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
 google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }
func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
 if File_google_ai_generativelanguage_v1beta_permission_proto != nil {
 return
 }
 file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
 }.Build()
 File_google_ai_generativelanguage_v1beta_permission_proto = out.File
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
 file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
 = nil
 file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//

```



```
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package discoveryengine_test

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
)

func ExampleUserLicenseClient_ListUserLicenses_all() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 for resp, err := range c.ListUserLicenses(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}
```

```

func ExampleUserLicenseClient_ListOperations_all() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
 }
 for resp, err := range c.ListOperations(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

```

// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 context "context"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
 protoreflect
 "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Assignment protocol for a license pool.
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

```

```

// The type of assignment protocol.
//
// Types that are assignable to AssignmentType:
//
// *AssignmentProtocol_ManualAssignmentType_
// *AssignmentProtocol_AutoAssignmentType_
AssignmentType
isAssignmentProtocol_AssignmentType `protobuf_oneof:"assignment_type"`
}

func (x *AssignmentProtocol) Reset() {
 *x = AssignmentProtocol{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol) ProtoMessage() {}

func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType {
 if m != nil {
 return m.AssignmentType
 }
}

```



```

return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {
if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
return x.ManualAssignmentType
}
return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
return x.AutoAssignmentType
}
return nil
}

type isAssignmentProtocol_AssignmentType interface {
isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_ struct {
// Allow manual assignments triggered by administrative operations only.
ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
`protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type
AssignmentProtocol_AutoAssignmentType_ struct {
// Allow automatic assignments triggered by data plane operations.
AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
`protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
}

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Identifier. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Required. Assignment protocol for the license pool.

```

```

LicenseAssignmentProtocol *AssignmentProtocol
`protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
json:"license_assignment_protocol,omitempty"`
// Output only. Licenses count that are available to be assigned.
AvailableLicenseCount int32
`protobuf:"varint,3,opt,name=available_license_count,json=availableLicenseCount,proto3"
json:"available_license_count,omitempty"`
// Output only. Total number of licenses in the pool.
TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total_license_count,json=totalLicenseCount,proto3"
json:"total_license_count,omitempty"`
}

func (x *LicensePool) Reset() {
 *x = LicensePool{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensePool) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensePool) ProtoMessage() {}

func (x *LicensePool) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.
func (*LicensePool) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{1}
}

func (x *LicensePool) GetName() string {

```

```

if x != nil {
 return x.Name
}
return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a
// license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

func (x *GetLicensePoolRequest) Reset() {
 *x = GetLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {

```

```

return protoimpl.X.MessageStringOf(x)
}

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if
 ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The license pool to update.
 //
 // The license pool's name field is used to identify the license pool to
 // update. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`.
 LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
 json:"license_pool,omitempty"`
}

```

```

Required. The list of fields to update.
UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
json:"update_mask,omitempty"`
}

func (x *UpdateLicensePoolRequest) Reset() {
 *x = UpdateLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UpdateLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UpdateLicensePoolRequest) ProtoMessage() {}

func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor
// instead.
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{3}
}

func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {
 if x != nil {
 return x.LicensePool
 }
 return nil
}

func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {

```

```

if x != nil {
 return x.UpdateMask
}
return nil
}

// Request message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func
(x *AssignRequest) Reset() {
 *x = AssignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{4}
}

func (x *AssignRequest) GetParent()
string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

```

```

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x
 != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{5}
}

// Request message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset()
{
 *x = UnassignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```



```

}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{6}
}

func (x *UnassignRequest) GetParent() string {
 if x != nil {
 return
 x.Parent
 }
 return ""
}

func (x *UnassignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

```

```

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//
// [LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
// ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. The maximum number of users to return. The service may return

```

```

// fewer than this value.
PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3"
json:"page_size,omitempty"`
// Optional. A page token, received from a previous `EnumerateLicensedUsers`
// call. Provide this to retrieve the subsequent page.
PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

func (x *EnumerateLicensedUsersRequest) Reset() {
*x = EnumerateLicensedUsersRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo()
== nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{8}
}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
if x != nil {
return x.Parent
}
return ""
}

```

```

}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

// A licensed user.
type LicensedUser struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Username.
 // Format: `name@domain.com`.
 Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
 //
 // Output only. Timestamp when the license was assigned.
 AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3" json:"assign_time,omitempty"`
 // Output only. Timestamp when the license was recently used. This may not be
 // the most recent usage time, and will be updated regularly (within 24
 // hours).
 RecentUsageTime *timestamppb.Timestamp
 `protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3" json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {
 *x = LicensedUser{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if
x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }
 return nil
}

// Response message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License

```

```

ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The list of licensed users.
 LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
 json:"licensed_users,omitempty"`
 // A token that can be sent as `page_token` to retrieve the next page.
 // If this field is omitted, there are no subsequent pages.
 NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
 json:"next_page_token,omitempty"`
}

func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersResponse) ProtoMessage() {}

func
(x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),

```

```

[]int{10}
}

func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
 return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 }
 return ms
}

```

```

}
return mi.MessageOf(x)
}

// Deprecated:
// Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Optional. The time to live for an inactive license. After this time has
 // passed, the license will be automatically unassigned from the user. Must
 // be at least 7 days, if set. If unset, the license will never expire.
 InactiveLicenseTtl *durationpb.Duration
 `protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
 json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 }
}

```



```

 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration {
 if x != nil {
 return x.InactiveLicenseTtl
 }
 return nil
}

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protorelect.FileDescriptor

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{
 0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
 0x6f,
 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
 0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
 0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,
 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,
 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x64,
 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,
 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,

```

0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,  
0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73,  
0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,  
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,  
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01,  
0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42, 0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,

0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02,

0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,  
0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69,  
0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,  
0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc3, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74,  
0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x89, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,

0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74,  
0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x6f, 0xda, 0x41, 0x18, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4e, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x3e, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,  
0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12, 0xdf, 0x01, 0x0a, 0x06,  
0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71,  
0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f,  
0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75,  
0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f, 0x3a, 0x01,  
0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62,  
0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a,  
0x2f, 0x6f, 0x72, 0x64,  
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0xe7, 0x01,  
0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73, 0x65, 0x22, 0x5a, 0xda, 0x41, 0x10,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73,  
0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31, 0x2f, 0x7b,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f,  
0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x75,

0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e, 0x75, 0x6d,  
0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,  
0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e,  
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74,  
0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31,  
0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22,  
0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4c,  
0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69,  
0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f,  
0x6f, 0x72, 0x64,  
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63, 0xca, 0x41,  
0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,  
0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75,  
0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72,  
0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,  
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63,  
0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2f,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70, 0x69, 0x76,  
0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0x3b,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa, 0x02, 0x2d,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x43, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x50,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x2d,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x5c, 0x50,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x32,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x43,  
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x3a, 0x3a,

```

0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
= file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f
unc() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
 })
 return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
 (*AssignmentProtocol)(nil),
 // 0: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 (*LicensePool)(nil), // 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
 (*GetLicensePoolRequest)(nil), // 2:
 google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 (*UpdateLicensePoolRequest)(nil), // 3:
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 (*AssignRequest)(nil), // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
 (*AssignResponse)(nil), // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
 (*UnassignRequest)(nil), // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 (*UnassignResponse)(nil), // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 (*EnumerateLicensedUsersRequest)(nil), // 8:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 (*LicensedUser)(nil),
 // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
 (*EnumerateLicensedUsersResponse)(nil), // 10:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
 (*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 (*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 (*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask

```

```

(*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
(*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =
[]int32{
 11, // 0:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name
 -> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12, // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name -
 > google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
 google.protobuf.FieldMask
 14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
 google.protobuf.Timestamp
 14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name
 -> google.protobuf.Timestamp
 9, // 7:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name ->
 google.cloud.commerce.consumer.procurement.v1.LicensedUser
 15, // 8:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
 ype_name -> google.protobuf.Duration
 2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
 > google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 3, // 10:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
 google.cloud.commerce.consumer.procurement.v1.AssignRequest
 6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type
 -> google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 8, // 13:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_typ
 e -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 1, // 14:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 1, // 15:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
 google.cloud.commerce.consumer.procurement.v1.AssignResponse
 7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
 google.cloud.commerce.consumer.procurement.v1.UnassignResponse

```



```

10, // 18:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_type
pe
-> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14, // [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}
func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File:
 protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
 }.Build()
 File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs
 = nil
}

```

```

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// LicenseManagementServiceClient is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {
 // Gets the license pool.
 GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error)
}

type licenseManagementServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
 return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool", in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

func
(c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest,
opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption)
(*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign", in,
out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

// LicenseManagementServiceServer is the server API for LicenseManagementService service.

```

type LicenseManagementServiceServer interface {
 // Gets the license pool.
 GetLicensePool(context.Context, *GetLicensePoolRequest) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(context.Context, *UpdateLicensePoolRequest) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(context.Context, *AssignRequest) (*AssignResponse,
 error)
 // Unassigns a license from a user.
 Unassign(context.Context, *UnassignRequest) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(context.Context, *EnumerateLicensedUsersRequest)
 (*EnumerateLicensedUsersResponse, error)
}

// UnimplementedLicenseManagementServiceServer can be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer struct {
}

func (*UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)
(*UnassignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest) (*EnumerateLicensedUsersResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

func RegisterLicenseManagementServiceServer(s *grpc.Server, srv LicenseManagementServiceServer) {
 s.RegisterService(&_LicenseManagementService_serviceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(GetLicensePoolRequest)

```

```

if err := dec(in); err != nil
{
 return nil, err
}
if interceptor == nil {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, in)
}
info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(UpdateLicensePoolRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool",
 }
 handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(AssignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Assign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,

```

```

FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv
interface{}, ctx context.Context, dec func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
 in := new(UnassignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(EnumerateLicensedUsersRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor
 == nil {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod:
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
 }
 return interceptor(ctx, in, info, handler)
}

```

```

}

var _LicenseManagementService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.commerce.consumer.procurement.v1.LicenseManagementService",
 HandlerType: (*LicenseManagementServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "GetLicensePool",
 Handler: _LicenseManagementService_GetLicensePool_Handler,
 },
 {
 MethodName: "UpdateLicensePool",
 Handler: _LicenseManagementService_UpdateLicensePool_Handler,
 },
 {
 MethodName: "Assign",
 Handler: _LicenseManagementService_Assign_Handler,
 },
 {
 MethodName: "Unassign",
 Handler: _LicenseManagementService_Unassign_Handler,
 },
 {
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

// [START discoveryengine_v1_generated_UserLicenseService_BatchUpdateUserLicenses_sync]

```

```

package main

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
)

func main() {
 ctx := context.Background()
 // This snippet has been automatically generated
 and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.BatchUpdateUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesReque
 st.
 }
 op, err := c.BatchUpdateUserLicenses(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 resp, err := op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

// [END discoveryengine_v1_generated_UserLicenseService_BatchUpdateUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.

```



```

// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/compute_pb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 // require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#DeleteLicenseRequest.
}
op, err := c.Delete(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {

```

```

ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
}
resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template
 only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#ListLicensesRequest.
 }
 it := c.List(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*computepb.LicensesListResponse)
 }
}

func
ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:

```

```

// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#SetIamPolicyLicenseRequest.
}
resp, err := c.SetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {

```

```

ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may
require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#UpdateLicenseRequest.
}
op, err := c.Update(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

```

# 1.101 prometheus-common 0.64.0

## 1.101.1 Available under license :

Common libraries shared by Prometheus Go components.  
 Copyright 2015 The Prometheus Authors

This product includes software developed at  
 SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
 and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,



excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.102 googlecloudgocomputemetadata 0.7.0

## 1.102.1 Available under license :

```
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

// [START discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]

package main

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 "google.golang.org/api/iterator"
)

func main() {
 ctx := context.Background()
 // This snippet
 has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
```

```

c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
}
it := c.ListUserLicenses(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove
 // this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
}

// [END discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

package procurement

```
import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"

 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport
 "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newLicenseManagementClientHook clientHook

// LicenseManagementCallOptions contains the retry settings for each method of LicenseManagementClient.
type LicenseManagementCallOptions struct {
 GetLicensePool []gax.CallOption
 UpdateLicensePool []gax.CallOption
 Assign []gax.CallOption
 Unassign []gax.CallOption
 EnumerateLicensedUsers []gax.CallOption
 GetOperation []gax.CallOption
}

func defaultLicenseManagementGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("cloudcommerceconsumerprocurement.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN:443"),
 ,
 internaloption.WithDefaultMTLSEndpoint("cloudcommerceconsumerprocurement.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 }
}
```

```

option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
}
}

func
defaultLicenseManagementCallOptions() *LicenseManagementCallOptions {
return &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
}
}

func defaultLicenseManagementRESTCallOptions() *LicenseManagementCallOptions {
return
&LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
}
}

// internalLicenseManagementClient is an interface that defines the methods available from Cloud Commerce
Consumer Procurement API.
type internalLicenseManagementClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 GetLicensePool(context.Context, *procurementpb.GetLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 UpdateLicensePool(context.Context, *procurementpb.UpdateLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 Assign(context.Context, *procurementpb.AssignRequest, ...gax.CallOption) (*procurementpb.AssignResponse,
error)
 Unassign(context.Context, *procurementpb.UnassignRequest,
...gax.CallOption) (*procurementpb.UnassignResponse, error)
 EnumerateLicensedUsers(context.Context, *procurementpb.EnumerateLicensedUsersRequest, ...gax.CallOption)
*LicensedUserIterator
 GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
}

```

```

// LicenseManagementClient is a client for interacting with Cloud Commerce Consumer Procurement API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// Service for managing licenses.
type LicenseManagementClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseManagementClient

 // The call options for this service.
 CallOptions *LicenseManagementCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseManagementClient)
 Close() error {
 return c.internalClient.Close()
 }

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseManagementClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseManagementClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// GetLicensePool gets the license pool.
func (c *LicenseManagementClient) GetLicensePool(ctx context.Context, req
 *procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 return c.internalClient.GetLicensePool(ctx, req, opts...)
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *LicenseManagementClient)
 UpdateLicensePool(ctx context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
 (*procurementpb.LicensePool, error) {

```

```

return c.internalClient.UpdateLicensePool(ctx, req, opts...)
}

// Assign assigns a license to a user.
func (c *LicenseManagementClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
return c.internalClient.Assign(ctx, req, opts...)
}

// Unassign unassigns a license from a user.
func (c *LicenseManagementClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest, opts
...gax.CallOption) (*procurementpb.UnassignResponse, error) {
return c.internalClient.Unassign(ctx, req, opts...)
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *LicenseManagementClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
return c.internalClient.EnumerateLicensedUsers(ctx,
req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *LicenseManagementClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
return c.internalClient.GetOperation(ctx, req, opts...)
}

// licenseManagementGRPCClient is a client for interacting with Cloud Commerce Consumer Procurement API
over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type licenseManagementGRPCClient struct {
// Connection pool of gRPC connections to the service.
connPool gtransport.ConnPool

// Points back to the CallOptions field of the containing LicenseManagementClient
CallOptions **LicenseManagementCallOptions

// The gRPC API client.
licenseManagementClient procurementpb.LicenseManagementServiceClient

operationsClient longrunningpb.OperationsClient

//
The x-goog-* metadata to be sent with each request.
xGoogHeaders []string

```



```

logger *slog.Logger
}

// NewLicenseManagementClient creates a new license management service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing licenses.
func NewLicenseManagementClient(ctx context.Context, opts ...option.ClientOption) (*LicenseManagementClient,
error) {
 clientOpts := defaultLicenseManagementGRPCClientOptions()
 if newLicenseManagementClientHook != nil {
 hookOpts, err := newLicenseManagementClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }
 client := LicenseManagementClient{CallOptions: defaultLicenseManagementCallOptions()}

 c := &licenseManagementGRPCClient{
 connPool:
 connPool,
 licenseManagementClient: procurementpb.NewLicenseManagementServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
 }
 c.setGoogleClientInfo()

 client.internalClient = c

 return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *licenseManagementGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

```

```

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc",
 grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 logger *slog.Logger
}

// NewLicenseManagementRESTClient creates a new license management service rest client.
//
// Service for managing licenses.
func NewLicenseManagementRESTClient(ctx context.Context,
 opts ...option.ClientOption) (*LicenseManagementClient, error) {
 clientOpts := append(defaultLicenseManagementRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseManagementRESTCallOptions()

```

```

c := &licenseManagementRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

return &LicenseManagementClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseManagementRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://cloudcommerceconsumerprocurement.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://cloudcommerceconsumerprocurement.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient
 = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.

```

```

func (c *licenseManagementRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *licenseManagementGRPCClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.GetLicensePool, req, settings.GRPC, c.logger,
"GetLicensePool")
 return err
 },
 opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.UpdateLicensePool, req, settings.GRPC, c.logger,
"UpdateLicensePool")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

func (c *licenseManagementGRPCClient)
Assign(ctx context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption)
(*procurementpb.AssignResponse, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 var resp *procurementpb.AssignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Assign, req, settings.GRPC, c.logger, "Assign")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 hds := []string{"x-goog-request-params",
 fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
 var resp *procurementpb.UnassignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Unassign, req, settings.GRPC, c.logger, "Unassign")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts

```

```

=
append((*c.CallOptions).EnumerateLicensedUsers[0:len((*c.CallOptions).EnumerateLicensedUsers):len((*c.CallOptions).EnumerateLicensedUsers)], opts...)
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.EnumerateLicensedUsers, req, settings.GRPC, c.logger,
"EnumerateLicensedUsers")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetLicensedUsers(), resp.GetNextPageToken(),
 nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *licenseManagementGRPCClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

// GetLicensePool gets the license pool.
func (c *licenseManagementRESTClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx,
hds...)
opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path

```

```

 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLicensePool")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *licenseManagementRESTClient) UpdateLicensePool(ctx
context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetLicensePool()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetLicensePool().GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 }

```



```

 }
 params.Add("updateMask", string(field[1:len(field)-1]))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Assign assigns a license to a user.

```

```

func
(c *licenseManagementRESTClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:assign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial:
true, DiscardUnknown: true}
 resp := &procurementpb.AssignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Assign")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }
 })
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Unassign unassigns a license from a user.
func (c *licenseManagementRESTClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err :=
 m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:unassign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &procurementpb.UnassignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
 err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 })
 if e != nil {
 return nil, e
 }
 return resp, nil
}

```

```

 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Unassign")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *licenseManagementRESTClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 it := &LicensedUserIterator{}
 req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if
 pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:enumerateLicensedUsers", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))

```

```

}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "EnumerateLicensedUsers")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)

```

```

it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetOperation is a utility method from
google.longrunning.Operations.
func (c *licenseManagementRESTClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }
 })
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/compute"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"

```

```

"google.golang.org/protobuf/proto"
)

var
newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
Delete []gax.CallOption
Get []gax.CallOption
GetIamPolicy []gax.CallOption
Insert []gax.CallOption
List []gax.CallOption
SetIamPolicy []gax.CallOption
TestIamPermissions []gax.CallOption
Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
return &LicensesCallOptions{
Delete: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier:
1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
GetIamPolicy: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
},

```



```

Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 TestIamPermissions(context.Context,
 *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
 Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

```

// LicensesClient is a client for interacting with Google Compute Engine API.

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with

```

method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.
 CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed
// on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context, req
 *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

```

```

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available in the
specified project. This method does not get any licenses that belong to other projects, including licenses attached to
publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to
make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

```

```

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type
licensesRESTClient struct {
// The http endpoint to connect to.
endpoint string

// The http client.
httpClient *http.Client

// operationClient is used to call the operation-specific management service.
operationClient *GlobalOperationsClient

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing LicensesClient
CallOptions **LicensesCallOptions

logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
if err != nil {
return nil, err
}

callOpts := defaultLicensesRESTCallOptions()
c := &licensesRESTClient{
endpoint: endpoint,
httpClient: httpClient,
CallOptions: &callOpts,
logger:
internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
option.WithHTTPClient(httpClient),
option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {

```

```

 return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
}
}

// setGoogleClientInfo sets the name and version of the application in
//
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
if err := c.operationClient.Close(); err != nil {
 return err
}
return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
return nil
}

```

```

// Delete deletes the specified license. Caution This resource is intended for use
// only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *compute.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &compute.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }
 })
}

```

```

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
 {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

```

```

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy",
 req.GetProject(), req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
 opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}

```



```

e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if
settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) Insert(ctx context.Context, req *compute.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

```

```

baseUrl, err := url.Parse(c.endpoint)
if
err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

```

```

params := url.Values{}

```

```

if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err
:= http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}

```

```

return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for

use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts

```

```

...gax.CallOption) *LicenseIterator {

```

```

it := &LicenseIterator{}

```

```

req = proto.Clone(req).(*computepb.ListLicensesRequest)

```

```

unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}

```

```

it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {

```

```

resp := &computepb.LicensesListResponse{}

```

```

if pageToken != "" {

```

```

 req.PageToken = proto.String(pageToken)

```

```

}

```

```

if pageSize > math.MaxInt32 {

```

```

 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))

```

```

} else if pageSize != 0 {

```

```

 req.MaxResults = proto.Uint32(uint32(pageSize))

```

```

}

```

```

baseUrl, err := url.Parse(c.endpoint)

```

```

if err != nil {

```

```

 return nil, "", err

```

```

}

```

```

baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

```

```

params := url.Values{}

```

```

if req != nil && req.Filter != nil {

```

```

 params.Add("filter",

```

```

 fmt.Sprintf("%v", req.GetFilter()))

```

```

}

```

```

if req != nil && req.MaxResults != nil {

```

```

 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))

```

```

}

```

```

if req != nil && req.OrderBy != nil {

```

```

 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))

```

```

}

```

```

if req != nil && req.PageToken != nil {

```

```

 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))

```

```

}

```

```

if req != nil && req.ReturnPartialSuccess != nil {

```

```

 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))

```

```

}

```

```

baseUrl.RawQuery = params.Encode()

```

```

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header
= headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

```

```

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace
images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {

```

```

 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) TestIamPermissions(ctx
context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)],
opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))

```

```

if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Update(ctx context.Context,
req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
 if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
 }
}

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &compute.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return
op, nil
}

```



## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

package compute

import (

"bytes"

"context"

"fmt"

"log/slog"

"net/http"

"net/url"

"time"

computepb "cloud.google.com/go/compute/apiv1beta/computepb"

gax "github.com/googleapis/gax-go/v2"

"google.golang.org/api/option"

"google.golang.org/api/option/internaloption"

```

httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
)

var
newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
 }
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.

```

```

//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return
 c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {

```

```

return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions

 logger *slog.Logger
}

// NewLicenseCodesRESTClient
// creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 }
}

```



```

internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
internaloption.WithDefaultUniverseDomain("googleapis.com"),
internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
internaloption.WithDefaultScopes(DefaultAuthScopes()...),
internaloption.EnableNewAuthLibrary(),
}
}

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
"x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
return
nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

```

```

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

hds
= append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.LicenseCode{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that
// a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)

```

```

if err != nil {
 return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v/testIamPermissions",
req.GetProject(), req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

```

```

}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/compute_pb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
)

var
 newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

```

```

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
return &LicenseCodesCallOptions{
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
}),
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
TestIamPermissions: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
Close()
error
setGoogleClientInfo(...string)
Connection() *grpc.ClientConn
Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
(*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
// The internal transport-dependent client.
internalClient internalLicenseCodesClient

// The call options for this service.
CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.

```

```

func (c *LicenseCodesClient) Close() error {
 return c.internalClient.Close()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

```

```

// Points back to the CallOptions field of the containing LicenseCodesClient
CallOptions **LicenseCodesCallOptions

logger *slog.Logger
}

// NewLicenseCodesRESTClient
creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)

```

```

kv = append(kv, "gax", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return
 nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.LicenseCode{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path

```



```

 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a
// caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v/testIamPermissions",
 req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

```

```

hds = append(c.xGoogHeaders, hds...)
hds =
append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

```
// See the License for the specific language governing permissions and
// limitations under the License.
```

```
// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1alpha/permission.proto
```

```
package generativelanguagepb
```

```
import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)
```

```
const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)
```

```
// Defines types of the grantee of this permission.
type Permission_GrantType int32
```

```
const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)
```

```
// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
```

```

"GROUP",
3: "EVERYONE",
}
Permission_GrantType_value = map[string]int32{
"GRANTEE_TYPE_UNSPECIFIED": 0,
"USER": 1,
"GROUP": 2,
"EVERYONE": 3,
}
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
p := new(Permission_GrantType)
*p = x
return p
}

func (x Permission_GrantType) String() string {
return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protorelect.EnumDescriptor {
return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protorelect.EnumType {
return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protorelect.EnumNumber {
return protorelect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
// The default value. This value is unused.
Permission_ROLE_UNSPECIFIED Permission_Role = 0
// Owner can use, update, share and delete the resource.
Permission_OWNER Permission_Role = 1
// Writer can use, update and share the resource.

```

```

Permission_WRITER Permission_Role = 2
// Reader can use the resource.
Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

```

```

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service
// accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user of group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.
 EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
 json:"email_address,omitempty"`
 // Required. The role granted by this permission.
 Role *Permission_Role
 `protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
 json:"role,omitempty"`
}

func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

```

```

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1alpha_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,
 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,
 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,
 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,
 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,
 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,
 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,
 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,
 0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,
 0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,
 0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,
 0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,
 0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,
 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,
 0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,
 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,
 0x4e,
 0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,
 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,
 0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,
 0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,

```



```

0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,
0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,
0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,
0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,
0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,
0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,
0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,
0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,
0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes
= []any{
 (Permission_GranteeType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GranteeType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
}
var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.GranteeType
 1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}
func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
 return
}
file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
= nil
file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}

```

```

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#ListLicensesRequest.
 }
 for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.

```

```

 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package discoveryengine_test

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewUserLicenseClient() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c

```

```

}

func ExampleNewUserLicenseRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseRESTClient(ctx)
 if
 err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleUserLicenseClient_BatchUpdateUserLicenses() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.BatchUpdateUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesReque
 st.
 }
 op, err := c.BatchUpdateUserLicenses(ctx, req)
 if err != nil {
 //
 TODO: Handle error.
 }

 resp, err := op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

}
// TODO: Use resp.
_ = resp
}

func ExampleUserLicenseClient_ListUserLicenses() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 it := c.ListUserLicenses(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
 }
}

func ExampleUserLicenseClient_CancelOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.

```

```

// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.CancelOperationRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#CancelOperationRequest.
}
err = c.CancelOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleUserLicenseClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }
 resp,
 err := c.GetOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleUserLicenseClient_ListOperations() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.

```

```

// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
}
it := c.ListOperations(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 //
 TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*longrunningpb.ListOperationsResponse)
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

```



```

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license_service.proto

package discoveryenginepb

import (
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 context "context"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 status "google.golang.org/genproto/googleapis/rpc/status"
 grpc
 "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status1 "google.golang.org/grpc/status"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Request message for
// [UserService.ListUserLicenses][google.cloud.discoveryengine.v1.UserService.ListUserLicenses].
type ListUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The parent [UserStore][] resource name,
 // format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 // Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. Requested page size. Server may return fewer items than
 // requested. If unspecified, defaults to 10. The maximum value is 50; values
 // above 50 will be coerced to 50.
 //
 // If this field is negative, an INVALID_ARGUMENT error is returned.

```

```

PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3" json:"page_size,omitempty"`
// Optional. A page token, received from a previous `ListUserLicenses` call.
// Provide this to retrieve the subsequent page.
//
// When paginating, all other parameters provided to `ListUserLicenses`
// must match the call that provided the page token.
PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
// Optional. Filter for the list request.
//
// Supported fields:
//
// * `license_assignment_state`
//
//
Examples:
//
// * `license_assignment_state = ASSIGNED` to list assigned user licenses.
// * `license_assignment_state = NO_LICENSE` to list not licensed users.
// * `license_assignment_state = NO_LICENSE_ATTEMPTED_LOGIN` to list users
// who attempted login but no license assigned.
// * `license_assignment_state != NO_LICENSE_ATTEMPTED_LOGIN` to filter
// out users who attempted login but no license assigned.
Filter string `protobuf:"bytes,4,opt,name=filter,proto3" json:"filter,omitempty"`
}

func (x *ListUserLicensesRequest) Reset() {
*x = ListUserLicensesRequest{}
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *ListUserLicensesRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*ListUserLicensesRequest) ProtoMessage() {}

func (x *ListUserLicensesRequest) ProtoReflect() protoreflect.Message {
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
if
x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
}

```

```

return mi.MessageOf(x)
}

// Deprecated: Use ListUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*ListUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{0}
}

func (x *ListUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *ListUserLicensesRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *ListUserLicensesRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

func (x *ListUserLicensesRequest) GetFilter() string {
 if x != nil {
 return x.Filter
 }
 return ""
}

// Response message for
// [UserLicenseService.ListUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses].
type
ListUserLicensesResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // All the customer's
 // [UserLicense][google.cloud.discoveryengine.v1.UserLicense]s.
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
}

```

```
// A token, which can be sent as `page_token` to retrieve the next page. If
// this field is omitted, there are no subsequent pages.
NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
json:"next_page_token,omitempty"`
}
```

```
func (x *ListUserLicensesResponse) Reset() {
 *x = ListUserLicensesResponse{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}
```

```
func (x *ListUserLicensesResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}
```

```
func
(*ListUserLicensesResponse) ProtoMessage() {}
```

```
func (x *ListUserLicensesResponse) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}
```

```
// Deprecated: Use ListUserLicensesResponse.ProtoReflect.Descriptor instead.
func (*ListUserLicensesResponse) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{1}
}
```

```
func (x *ListUserLicensesResponse) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}
```

```
func (x *ListUserLicensesResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
}
```

```

return ""
}

// Request message for
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdat
eUserLicenses]
//
method.
type BatchUpdateUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The source of the input.
 //
 // Types that are assignable to Source:
 //
 // *BatchUpdateUserLicensesRequest_InlineSource_
 Source isBatchUpdateUserLicensesRequest_Source `protobuf_oneof:"source"`
 // Required. The parent [UserStore][] resource name, format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. If true, if user licenses removed associated license config, the
 // user license will be deleted. By default which is false, the user license
 // will be updated to unassigned state.
 DeleteUnassignedUserLicenses bool
 `protobuf:"varint,4,opt,name=delete_unassigned_user_licenses,json=deleteUnassignedUserLicenses,proto3"
 json:"delete_unassigned_user_licenses,omitempty"`
}

func
(x *BatchUpdateUserLicensesRequest) Reset() {
 *x = BatchUpdateUserLicensesRequest{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest) ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
 if x != nil {

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(),
 []int{2}
}

func (m *BatchUpdateUserLicensesRequest) GetSource() isBatchUpdateUserLicensesRequest_Source {
 if m != nil {
 return m.Source
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest) GetInlineSource() *BatchUpdateUserLicensesRequest_InlineSource {
 if x, ok := x.GetSource().(*BatchUpdateUserLicensesRequest_InlineSource_); ok {
 return x.InlineSource
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *BatchUpdateUserLicensesRequest) GetDeleteUnassignedUserLicenses() bool {
 if x != nil {
 return x.DeleteUnassignedUserLicenses
 }
 return false
}

type isBatchUpdateUserLicensesRequest_Source interface {
 isBatchUpdateUserLicensesRequest_Source()
}

type BatchUpdateUserLicensesRequest_InlineSource_ struct {

```

```

// The inline source for the input content for document embeddings.
InlineSource *BatchUpdateUserLicensesRequest_InlineSource
`protobuf:"bytes,2,opt,name=inline_source,json=inlineSource,proto3,oneof"`
}

func
(*BatchUpdateUserLicensesRequest_InlineSource_) isBatchUpdateUserLicensesRequest_Source() {}

// Metadata related to the progress of the
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses]
// operation. This will be returned by the google.longrunning.Operation.metadata
// field.
type BatchUpdateUserLicensesMetadata struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Operation create time.
 CreateTime *timestamppb.Timestamp `protobuf:"bytes,1,opt,name=create_time,json=createTime,proto3"
 json:"create_time,omitempty"`
 // Operation last update time. If the operation is done, this is also the
 // finish time.
 UpdateTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=update_time,json=updateTime,proto3"
 json:"update_time,omitempty"`
 // Count of user licenses successfully updated.
 SuccessCount
 int64 `protobuf:"varint,3,opt,name=success_count,json=successCount,proto3" json:"success_count,omitempty"`
 // Count of user licenses that failed to be updated.
 FailureCount int64 `protobuf:"varint,4,opt,name=failure_count,json=failureCount,proto3"
 json:"failure_count,omitempty"`
}

func (x *BatchUpdateUserLicensesMetadata) Reset() {
 *x = BatchUpdateUserLicensesMetadata{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesMetadata) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesMetadata) ProtoMessage() {}

func (x *BatchUpdateUserLicensesMetadata) ProtoReflect() protoreflect.Message {

```

```

mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return
ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesMetadata.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesMetadata) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{3}
}

func (x *BatchUpdateUserLicensesMetadata) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetSuccessCount() int64 {
 if x != nil {
 return x.SuccessCount
 }
 return 0
}

func (x *BatchUpdateUserLicensesMetadata) GetFailureCount() int64 {
 if x != nil {
 return x.FailureCount
 }
 return 0
}

// Response message for
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdat

```



```

eUserLicenses]
// method.
type
BatchUpdateUserLicensesResponse struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// UserLicenses successfully updated.
UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
json:"user_licenses,omitempty"`
// A sample of errors encountered while processing the request.
ErrorSamples []*status.Status `protobuf:"bytes,2,rep,name=error_samples,json=errorSamples,proto3"
json:"error_samples,omitempty"`
}

func (x *BatchUpdateUserLicensesResponse) Reset() {
*x = BatchUpdateUserLicensesResponse{}
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesResponse) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesResponse) ProtoMessage() {}

func (x *BatchUpdateUserLicensesResponse) ProtoReflect() protoreflect.Message {
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesResponse.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesResponse) Descriptor() ([]byte, []int) {
return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{4}
}

func (x *BatchUpdateUserLicensesResponse) GetUserLicenses() []*UserLicense {

```

```

if x != nil {
 return x.UserLicenses
}
return nil
}

func (x *BatchUpdateUserLicensesResponse) GetErrorSamples() []*status.Status {
 if x != nil {
 return x.ErrorSamples
 }
 return nil
}

// The inline source for the input config for BatchUpdateUserLicenses
// method.
type BatchUpdateUserLicensesRequest_InlineSource struct {
 state protoimpl.MessageState
 sizeCache
 protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. A list of user licenses to update. Each user license
 // must have a valid
 // [UserLicense.user_principal][google.cloud.discoveryengine.v1.UserLicense.user_principal].
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
 // Optional. The list of fields to update.
 UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
 json:"update_mask,omitempty"`
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) Reset() {
 *x = BatchUpdateUserLicensesRequest_InlineSource{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest_InlineSource)
 ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest_InlineSource) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]
 if x != nil {

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest_InlineSource.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest_InlineSource) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{2, 0}
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

var File_google_cloud_discoveryengine_v1_user_license_service_proto
 protoreflect.FileDescriptor

var file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc = []byte{
 0x0a, 0x3a, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,
 0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x73,
 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f,
 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f,
 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1c, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f, 0x74, 0x61,
 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e,
 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,
 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x1a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,

```

0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x6c, 0x6f, 0x6e,  
0x67, 0x72, 0x75, 0x6e, 0x6e, 0x69, 0x6e, 0x67, 0x2f, 0x6f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69,  
0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2f, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64,  
0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x72, 0x70, 0x63, 0x2f, 0x73, 0x74, 0x61, 0x74, 0x75, 0x73, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc6, 0x01, 0x0a, 0x17, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x12, 0x48, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x30, 0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f,  
0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f,  
0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74,  
0x6f, 0x72, 0x65, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70,  
0x61, 0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a,  
0x0a, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65,  
0x6e, 0x12, 0x1b, 0x0a, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x22, 0x95,  
0x01, 0x0a, 0x18, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75,  
0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18,  
0x01, 0x20, 0x03,  
0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e,  
0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x52, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x26,  
0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65,  
0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65, 0x78, 0x74, 0x50, 0x61, 0x67,  
0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x22, 0xe0, 0x03, 0x0a, 0x1e, 0x42, 0x61, 0x74, 0x63, 0x68,  
0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x73, 0x0a, 0x0d, 0x69, 0x6e, 0x6c,  
0x69, 0x6e, 0x65, 0x5f, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b,  
0x32, 0x4c,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,  
0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x2e, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x48, 0x00,  
0x52, 0x0c, 0x69, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x48,  
0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x30,  
0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,  
0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70,

0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x12, 0x4a, 0x0a, 0x1f, 0x64, 0x65, 0x6c, 0x65,  
0x74, 0x65, 0x5f, 0x75, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65, 0x64, 0x5f, 0x75, 0x73,  
0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x08, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x1c, 0x64, 0x65, 0x6c, 0x65, 0x74, 0x65, 0x55, 0x6e,  
0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x73, 0x1a, 0xa8, 0x01, 0x0a, 0x0c, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53,  
0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x56, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x42, 0x03,  
0xe0, 0x41, 0x02, 0x52,  
0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x40, 0x0a,  
0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73, 0x6b, 0x42,  
0x08, 0x0a, 0x06, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x22, 0xe5, 0x01, 0x0a, 0x1f, 0x42, 0x61,  
0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63,  
0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d, 0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x12, 0x3b, 0x0a,  
0x0b, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f,  
0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a,  
0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x3b, 0x0a, 0x0b, 0x75, 0x70,  
0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32,  
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75,  
0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a, 0x75, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x23, 0x0a, 0x0d, 0x73, 0x75, 0x63, 0x63, 0x65,  
0x73, 0x73, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x03, 0x52, 0x0c,  
0x73, 0x75, 0x63, 0x63, 0x65, 0x73, 0x73, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x23, 0x0a, 0x0d,  
0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x04, 0x20,  
0x01, 0x28, 0x03, 0x52, 0x0c, 0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x22, 0xad, 0x01, 0x0a, 0x1f,  
0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74,  
0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x73,  
0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x52, 0x0c, 0x75, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x37, 0x0a, 0x0d, 0x65, 0x72, 0x72, 0x6f,  
0x72, 0x5f, 0x73, 0x61, 0x6d, 0x70, 0x6c, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x0b, 0x32,  
0x12, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x72, 0x70, 0x63, 0x2e, 0x53, 0x74, 0x61,  
0x74, 0x75, 0x73, 0x52, 0x0c, 0x65, 0x72, 0x72, 0x6f, 0x72, 0x53,

0x61, 0x6d, 0x70, 0x6c, 0x65,  
0x73, 0x32, 0x9a, 0x05, 0x0a, 0x12, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd7, 0x01, 0x0a, 0x10, 0x4c, 0x69, 0x73,  
0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x38, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73,  
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e,  
0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x39, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,  
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73,  
0x65, 0x22, 0x4e, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4,  
0x93, 0x02, 0x3f, 0x12, 0x3d, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x3d, 0x70, 0x72, 0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61,  
0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72,  
0x65, 0x73, 0x2f, 0x2a, 0x7d, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x12, 0xd5, 0x02, 0x0a, 0x17, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x3f,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69,  
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31,  
0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a,  
0x1d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x6c, 0x6f, 0x6e, 0x67, 0x72, 0x75, 0x6e,  
0x6e, 0x69, 0x6e, 0x67, 0x2e, 0x4f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x22, 0xd9,  
0x01, 0xca, 0x41, 0x82, 0x01, 0x0a, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,  
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65,  
0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,  
0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d,  
0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4d, 0x3a, 0x01, 0x2a,  
0x22, 0x48, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x70, 0x72,  
0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61, 0x74, 0x69, 0x6f,  
0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65, 0x73, 0x2f,  
0x2a, 0x7d, 0x3a, 0x62, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x1a, 0x52, 0xca, 0x41, 0x1e, 0x64,  
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0xd2, 0x41, 0x2e,  
0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67, 0x6f, 0x6f, 0x67,  
0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75, 0x74,  
0x68, 0x2f,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72, 0x6d, 0x42, 0x8a,  
0x02, 0x0a, 0x23, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,

```

0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x42, 0x17, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,
0x6e, 0x73, 0x65, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50,
0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64,
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62,
0xa2, 0x02, 0x0f,
0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45, 0x52, 0x59, 0x45, 0x4e, 0x47, 0x49,
0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75,
0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e,
0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c,
0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a,
0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc
)

func
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce.Do(func() {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData
}

var file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 6)
var file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = []any{
 (*ListUserLicensesRequest)(nil), // 0: google.cloud.discoveryengine.v1.ListUserLicensesRequest
 (*ListUserLicensesResponse)(nil), // 1: google.cloud.discoveryengine.v1.ListUserLicensesResponse
 (*BatchUpdateUserLicensesRequest)(nil), // 2:
 google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
 (*BatchUpdateUserLicensesMetadata)(nil), // 3: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata
 (*BatchUpdateUserLicensesResponse)(nil), // 4:
 google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse
 (*BatchUpdateUserLicensesRequest_InlineSource)(nil), // 5:
 google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
}

```

```

(*UserLicense)(nil), // 6: google.cloud.discoveryengine.v1.UserLicense
(*timestamppb.Timestamp)(nil), // 7: google.protobuf.Timestamp
(*status.Status)(nil), // 8: google.rpc.Status
(*fieldmaskpb.FieldMask)(nil), // 9: google.protobuf.FieldMask
(*longrunningpb.Operation)(nil), // 10: google.longrunning.Operation
}
var file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = []int32{
 6, // 0: google.cloud.discoveryengine.v1.ListUserLicensesResponse.user_licenses:type_name -> google.cloud.discoveryengine.v1.UserLicense
 5, // 1: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.inline_source:type_name -> google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
 7, // 2: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.create_time:type_name -> google.protobuf.Timestamp
 7, // 3: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.update_time:type_name -> google.protobuf.Timestamp
 6, // 4: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.user_licenses:type_name -> google.cloud.discoveryengine.v1.UserLicense
 8, // 5: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.error_samples:type_name -> google.rpc.Status
 6, // 6: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.user_licenses:type_name -> google.cloud.discoveryengine.v1.UserLicense
 9, // 7: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.update_mask:type_name -> google.protobuf.FieldMask
 0, // 8: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:input_type -> google.cloud.discoveryengine.v1.ListUserLicensesRequest
 2, // 9: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:input_type -> google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
 1, // 10: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:output_type -> google.cloud.discoveryengine.v1.ListUserLicensesResponse
 10, // 11: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:output_type -> google.longrunning.Operation
 10, // [10:12] is the sub-list for method output_type
 8, // [8:10] is the sub-list for method input_type
 8, // [8:8] is the sub-list for extension type_name
 8, // [8:8] is the sub-list for extension extendee
 0, // [0:8] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_service_proto_init()
}
func file_google_cloud_discoveryengine_v1_user_license_service_proto_init() {
 if File_google_cloud_discoveryengine_v1_user_license_service_proto != nil {
 return
 }
 file_google_cloud_discoveryengine_v1_user_license_proto_init()
 file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2].OneofWrappers = []any{
 (*BatchUpdateUserLicensesRequest_InlineSource)(nil),
 }
}

```



```

type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 6,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes: file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes,
 DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs,
 MessageInfos: file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes,
}.Build()
File_google_cloud_discoveryengine_v1_user_license_service_proto
= out.File
file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc = nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = nil
}

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// UserLicenseServiceClient is the client API for UserLicenseService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type UserLicenseServiceClient interface {
 // Lists the User Licenses.
 ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest,
 opts ...grpc.CallOption) (*ListUserLicensesResponse, error)
 // Updates the User License.
 // This method is used for batch assign/unassign licenses to users.
 BatchUpdateUserLicenses(ctx context.Context, in *BatchUpdateUserLicensesRequest, opts ...grpc.CallOption)
 (*longrunningpb.Operation, error)
}

type userLicenseServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewUserLicenseServiceClient(cc grpc.ClientConnInterface) UserLicenseServiceClient {

```

```

return &userLicenseServiceClient{cc}
}

func (c *userLicenseServiceClient) ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest, opts
...grpc.CallOption) (*ListUserLicensesResponse, error) {
 out := new(ListUserLicensesResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.discoveryengine.v1.UserLicenseService/ListUserLicenses", in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *userLicenseServiceClient) BatchUpdateUserLicenses(ctx context.Context, in
*BatchUpdateUserLicensesRequest,
opts ...grpc.CallOption) (*longrunningpb.Operation, error) {
 out := new(longrunningpb.Operation)
 err := c.cc.Invoke(ctx, "/google.cloud.discoveryengine.v1.UserLicenseService/BatchUpdateUserLicenses", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

// UserLicenseServiceServer is the server API for UserLicenseService service.
type UserLicenseServiceServer interface {
 // Lists the User Licenses.
 ListUserLicenses(context.Context, *ListUserLicensesRequest) (*ListUserLicensesResponse, error)
 // Updates the User License.
 // This method is used for batch assign/unassign licenses to users.
 BatchUpdateUserLicenses(context.Context, *BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error)
}

// UnimplementedUserLicenseServiceServer can be embedded to have forward compatible implementations.
type UnimplementedUserLicenseServiceServer struct {
}

func (*UnimplementedUserLicenseServiceServer) ListUserLicenses(context.Context, *ListUserLicensesRequest)
(*ListUserLicensesResponse, error) {
 return nil, status1.Errorf(codes.Unimplemented, "method ListUserLicenses not implemented")
}

func (*UnimplementedUserLicenseServiceServer) BatchUpdateUserLicenses(context.Context,
*BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error) {
 return nil, status1.Errorf(codes.Unimplemented, "method BatchUpdateUserLicenses not implemented")
}

func RegisterUserLicenseServiceServer(s *grpc.Server, srv UserLicenseServiceServer) {

```

```

s.RegisterService(&_UserLicenseService_serviceDesc, srv)
}

func _UserLicenseService_ListUserLicenses_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(ListUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.discoveryengine.v1.UserLicenseService/ListUserLicenses",
 }
 handler
 := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, req.(*ListUserLicensesRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _UserLicenseService_BatchUpdateUserLicenses_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(BatchUpdateUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.discoveryengine.v1.UserLicenseService/BatchUpdateUserLicenses",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, req.(*BatchUpdateUserLicensesRequest))
 }
 return interceptor(ctx,
 in, info, handler)
}

var _UserLicenseService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.discoveryengine.v1.UserLicenseService",
 HandlerType: (*UserLicenseServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {

```

```

 MethodName: "ListUserLicenses",
 Handler: _UserService_ListUserLicenses_Handler,
 },
 {
 MethodName: "BatchUpdateUserLicenses",
 Handler: _UserService_BatchUpdateUserLicenses_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/discoveryengine/v1/user_license_service.proto",
}

// Copyright 2023 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"

```

)

```
func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO:
 // Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
```

```

// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicenseCodesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#TestIamPermissionsLicenseCodeRequest.
}
resp,
err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2024 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

```

// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/compute_pb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute_pb#ListLicensesRequest.
 }
 for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//

```

```

// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license.proto

package discoveryenginepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect
 "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// License assignment state enumeration.
type UserLicense_LicenseAssignmentState int32

const (
 // Default value.
 UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED UserLicense_LicenseAssignmentState = 0
 // License assigned to the user.
 UserLicense_ASSIGNED UserLicense_LicenseAssignmentState = 1
 // No license assigned to the user.
 // Deprecated, translated to NO_LICENSE.
 UserLicense_UNASSIGNED UserLicense_LicenseAssignmentState = 2
 // No license assigned to the user.
 UserLicense_NO_LICENSE UserLicense_LicenseAssignmentState = 3
 // User attempted to login but no license assigned to the user.
 // This state is only used for no user first time login attempt

```



```

but cannot
// get license assigned.
// Users already logged in but cannot get license assigned will be assigned
// NO_LICENSE state(License could be unassigned by admin).
UserLicense_NO_LICENSE_ATTEMPTED_LOGIN UserLicense_LicenseAssignmentState = 4
)

// Enum value maps for UserLicense_LicenseAssignmentState.
var (
 UserLicense_LicenseAssignmentState_name = map[int32]string{
 0: "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED",
 1: "ASSIGNED",
 2: "UNASSIGNED",
 3: "NO_LICENSE",
 4: "NO_LICENSE_ATTEMPTED_LOGIN",
 }
 UserLicense_LicenseAssignmentState_value = map[string]int32{
 "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED": 0,
 "ASSIGNED": 1,
 "UNASSIGNED": 2,
 "NO_LICENSE": 3,
 "NO_LICENSE_ATTEMPTED_LOGIN": 4,
 }
)

func (x UserLicense_LicenseAssignmentState) Enum() *UserLicense_LicenseAssignmentState {
 p := new(UserLicense_LicenseAssignmentState)
 *p = x
 return p
}

func (x UserLicense_LicenseAssignmentState)
String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

func (UserLicense_LicenseAssignmentState) Descriptor() protorelect.EnumDescriptor {
 return file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0].Descriptor()
}

func (UserLicense_LicenseAssignmentState) Type() protorelect.EnumType {
 return &file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0]
}

func (x UserLicense_LicenseAssignmentState) Number() protorelect.EnumNumber {
 return protorelect.EnumNumber(x)
}

```

```

// Deprecated: Use UserLicense_LicenseAssignmentState.Descriptor instead.
func (UserLicense_LicenseAssignmentState) EnumDescriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0, 0}
}

// User License information assigned by the admin.
type UserLicense struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 //
 // Required. Immutable. The user principal of the User, could be email address
 // or other principal identifier. This field is immutable. Admin assign
 // licenses based on the user principal.
 UserPrincipal string `protobuf:"bytes,1,opt,name=user_principal,json=userPrincipal,proto3"
 json:"user_principal,omitempty"`
 // Optional. The user profile.
 // We use user full name(First name + Last name) as user profile.
 UserProfile string `protobuf:"bytes,3,opt,name=user_profile,json=userProfile,proto3"
 json:"user_profile,omitempty"`
 // Output only. License assignment state of the user.
 // If the user is assigned with a license config, the user login will be
 // assigned with the license;
 // If the user's license assignment state is unassigned or unspecified, no
 // license config will be associated to the user;
 LicenseAssignmentState UserLicense_LicenseAssignmentState
 `protobuf:"varint,4,opt,name=license_assignment_state,json=licenseAssignmentState,proto3,enum=google.cloud.discoveryengine.v1.UserLicense_LicenseAssignmentState"
 json:"license_assignment_state,omitempty"`
 // Optional. The full resource name of the Subscription(LicenseConfig)
 // assigned to the user.
 LicenseConfig string `protobuf:"bytes,5,opt,name=license_config,json=licenseConfig,proto3"
 json:"license_config,omitempty"`
 // Output only. User created timestamp.
 CreateTime *timestamppb.Timestamp `protobuf:"bytes,6,opt,name=create_time,json=createTime,proto3"
 json:"create_time,omitempty"`
 // Output only. User update timestamp.
 UpdateTime *timestamppb.Timestamp `protobuf:"bytes,7,opt,name=update_time,json=updateTime,proto3"
 json:"update_time,omitempty"`
 // Output only. User last logged in time.
 // If the user has not logged in yet, this field will be empty.
 LastLoginTime *timestamppb.Timestamp
 `protobuf:"bytes,8,opt,name=last_login_time,json=lastLoginTime,proto3" json:"last_login_time,omitempty"`
}

func (x *UserLicense) Reset() {
 *x = UserLicense{}
}

```

```

mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
ms
:= protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *UserLicense) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UserLicense) ProtoMessage() {}

func (x *UserLicense) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UserLicense.ProtoReflect.Descriptor instead.
func (*UserLicense) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0}
}

func (x *UserLicense) GetUserPrincipal() string {
 if x != nil {
 return x.UserPrincipal
 }
 return ""
}

func (x *UserLicense) GetUserProfile() string {
 if x != nil {
 return x.UserProfile
 }
 return ""
}

func (x *UserLicense) GetLicenseAssignmentState()
UserLicense_LicenseAssignmentState {
 if x != nil {
 return x.LicenseAssignmentState
 }
}

```

```
 return UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED
}
```

```
func (x *UserLicense) GetLicenseConfig() string {
 if x != nil {
 return x.LicenseConfig
 }
 return ""
}
```

```
func (x *UserLicense) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}
```

```
func (x *UserLicense) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}
```

```
func (x *UserLicense) GetLastLoginTime() *timestamppb.Timestamp {
 if x != nil {
 return x.LastLoginTime
 }
 return nil
}
```

```
var File_google_cloud_discoveryengine_v1_user_license_proto protoreflect.FileDescriptor
```

```
var file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = []byte{
 0x0a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,
 0x31,
 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f,
 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,
 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,
 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,
 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,
 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x22, 0xa6, 0x05, 0x0a, 0x0b, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,
 0x73, 0x65, 0x12, 0x2d, 0x0a, 0x0e,
```

0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x69, 0x6e, 0x63,  
0x69, 0x70, 0x61, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x05, 0xe0,  
0x41, 0x02, 0x52, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x50, 0x72, 0x69, 0x6e, 0x63, 0x69, 0x70, 0x61,  
0x6c, 0x12, 0x26, 0x0a, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x6f, 0x66, 0x69, 0x6c,  
0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x0b, 0x75, 0x73,  
0x65, 0x72, 0x50, 0x72, 0x6f, 0x66, 0x69, 0x6c, 0x65, 0x12, 0x82, 0x01, 0x0a, 0x18, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74,  
0x5f, 0x73, 0x74, 0x61, 0x74, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x43, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e,  
0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74,  
0x65, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x16, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74, 0x65, 0x12, 0x5b,  
0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x6e, 0x66, 0x69, 0x67,  
0x18, 0x05, 0x20, 0x01, 0x28, 0x09, 0x42, 0x34, 0xe0, 0x41, 0x01, 0xfa, 0x41, 0x2e, 0x0a, 0x2c,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x52, 0x0d, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x12, 0x40, 0x0a, 0x0b, 0x63,  
0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x06, 0x20, 0x01, 0x28, 0x0b,  
0x32,  
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,  
0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41,  
0x03, 0x52, 0x0a, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x07, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x47, 0x0a, 0x0f, 0x6c, 0x61, 0x73, 0x74, 0x5f, 0x6c, 0x6f, 0x67, 0x69, 0x6e, 0x5f, 0x74, 0x69,  
0x6d, 0x65, 0x18, 0x08, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73,  
0x74, 0x61, 0x6d, 0x70, 0x42,  
0x03, 0xe0, 0x41, 0x03, 0x52, 0x0d, 0x6c, 0x61, 0x73, 0x74, 0x4c,  
0x6f, 0x67, 0x69, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x22, 0x90, 0x01, 0x0a, 0x16, 0x4c, 0x69, 0x63,  
0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74,  
0x61, 0x74, 0x65, 0x12, 0x28, 0x0a, 0x24, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41,  
0x53, 0x53, 0x49, 0x47, 0x4e, 0x4d, 0x45, 0x4e, 0x54, 0x5f, 0x53, 0x54, 0x41, 0x54, 0x45, 0x5f,  
0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x0c, 0x0a,  
0x08, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x01, 0x12, 0x0e, 0x0a, 0x0a, 0x55,  
0x4e, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x02, 0x12, 0x0e, 0x0a, 0x0a, 0x4e,  
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x10, 0x03, 0x12, 0x1e, 0x0a, 0x1a, 0x4e,  
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41, 0x54, 0x54, 0x45, 0x4d, 0x50,  
0x54, 0x45, 0x44, 0x5f, 0x4c, 0x4f, 0x47, 0x49, 0x4e, 0x10,  
0x04, 0x42, 0x83, 0x02, 0x0a, 0x23,  
0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65,

```

0x2e, 0x76, 0x31, 0x42, 0x10, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69,
0x76, 0x31, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,
0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,
0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0xa2, 0x02, 0x0f, 0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45,
0x52, 0x59, 0x45, 0x4e, 0x47, 0x49, 0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67,
0x6c,
0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f,
0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76,
0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69,
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56,
0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc
)

func file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce.Do(func()
 {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData
}

var file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes = make([]protoimpl.EnumInfo, 1)
var file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = []any{
 (UserLicense_LicenseAssignmentState)(0), // 0:
 google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 (*UserLicense)(nil), // 1: google.cloud.discoveryengine.v1.UserLicense
 (*timestamppb.Timestamp)(nil), // 2: google.protobuf.Timestamp
}
var file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = []int32{
 0, // 0: google.cloud.discoveryengine.v1.UserLicense.license_assignment_state:type_name
 -> google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 2, // 1: google.cloud.discoveryengine.v1.UserLicense.create_time:type_name -> google.protobuf.Timestamp
 2, // 2: google.cloud.discoveryengine.v1.UserLicense.update_time:type_name -> google.protobuf.Timestamp
 2, // 3: google.cloud.discoveryengine.v1.UserLicense.last_login_time:type_name -> google.protobuf.Timestamp
 4, // [4:4] is the sub-list for method output_type

```

```

4, // [4:4] is the sub-list for method input_type
4, // [4:4] is the sub-list for extension type_name
4, // [4:4] is the sub-list for extension extendee
0, // [0:4] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_proto_init() }
func file_google_cloud_discoveryengine_v1_user_license_proto_init() {
if File_google_cloud_discoveryengine_v1_user_license_proto != nil {
 return
}
type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc,
 NumEnums: 1,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_cloud_discoveryengine_v1_user_license_proto_goTypes,
 DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs,
 EnumInfos: file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes,
 MessageInfos: file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes,
}.Build()
File_google_cloud_discoveryengine_v1_user_license_proto = out.File
file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = nil
file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = nil
file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

```

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := procurement.NewLicenseManagementRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseManagementClient_Assign() {

```



```

ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
}
resp, err := c.Assign(ctx, req)
if err != nil {
 // TODO: Handle error.
}
//
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicensedUsersRequest.
 }
 it := c.EnumerateLicensedUsers(ctx, req)
 for {
 resp, err := it.Next()

```

```

if err == iterator.Done {
 break
}
if err != nil {
 // TODO: Handle error.
}
//
TODO: Use resp.
_ = resp

// If you need to access the underlying RPC response,
// you can do so by casting the `Response` as below.
// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
}
}

func ExampleLicenseManagementClient_GetLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRequest.
 }
 resp, err := c.GetLicensePool(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_Unassign() {
 ctx := context.Background()

```

```

// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.UnassignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest.
}
resp,
err := c.Unassign(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_UpdateLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UpdateLicensePoolRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
 }
 resp, err := c.UpdateLicensePool(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```
//
TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }
 resp, err := c.GetOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

//go:build go1.23

package compute\_test

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

package compute\_test

import (

"context"

```

compute "cloud.google.com/go/compute/apiv1beta"
computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
"google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 //
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute_pb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute_pb#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetIamPolicyLicenseRequest.
 }

```

```

resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded
 // as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```



```

}
defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#ListLicensesRequest.
}
it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _
 = it.Response.(*computepb.LicensesListResponse)
}
}

func ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#SetIamPolicyLicenseRequest.
 }
 resp, err := c.SetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

```

```

}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#UpdateLicenseRequest.

```

```

}
op, err := c.Update(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/compute_pb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request
 struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseCodeRequest.

```

```

}
resp,
err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
)

func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()
 // This snippet has been automatically
 // generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

defer c.Close()

req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
}
for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package discoveryengine

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 "cloud.google.com/go/longrunning"
 lroauto "cloud.google.com/go/longrunning/autogen"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"

```

```

gax
"github.com/googleapis/gax-go/v2"
"google.golang.org/api/iterator"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
gtransport "google.golang.org/api/transport/grpc"
httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/grpc/codes"
"google.golang.org/protobuf/encoding/protojson"
"google.golang.org/protobuf/proto"
)

var newUserLicenseClientHook clientHook

// UserLicenseCallOptions contains the retry settings for each method of UserLicenseClient.
type UserLicenseCallOptions struct {
 ListUserLicenses []gax.CallOption
 BatchUpdateUserLicenses []gax.CallOption
 CancelOperation []gax.CallOption
 GetOperation []gax.CallOption
 ListOperations []gax.CallOption
}

func defaultUserLicenseGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("discoveryengine.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("discoveryengine.UNIVERSE_DOMAIN:443"),
 internaloption.WithDefaultMTLSEndpoint("discoveryengine.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
 }
}

func
defaultUserLicenseCallOptions() *UserLicenseCallOptions {
 return &UserLicenseCallOptions{
 ListUserLicenses: []gax.CallOption{},
 BatchUpdateUserLicenses: []gax.CallOption{},
 CancelOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{

```

```

 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier:
1.30,
 })
 }),
},
GetOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
ListOperations: []gax.CallOption{
 gax.WithTimeout(300000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 30000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
}
}

func defaultUserLicenseRESTCallOptions() *UserLicenseCallOptions {
 return &UserLicenseCallOptions{
 ListUserLicenses: []gax.CallOption{},
 BatchUpdateUserLicenses: []gax.CallOption{},
 CancelOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer
{
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 })
 }
}

```



```

 },
 http.StatusServiceUnavailable)
)),
},
GetOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusServiceUnavailable)
)),
},
ListOperations: []gax.CallOption{
 gax.WithTimeout(300000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 30000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusServiceUnavailable)
)),
},
}
}

```

// internalUserLicenseClient is an interface that defines the methods available from Discovery Engine API.

type internalUserLicenseClient

interface {

Close() error

setGoogleClientInfo(...string)

Connection() \*grpc.ClientConn

ListUserLicenses(context.Context, \*discoveryenginepb.ListUserLicensesRequest, ...gax.CallOption)

\*UserLicenseIterator

BatchUpdateUserLicenses(context.Context, \*discoveryenginepb.BatchUpdateUserLicensesRequest,

...gax.CallOption) (\*BatchUpdateUserLicensesOperation, error)

BatchUpdateUserLicensesOperation(name string) \*BatchUpdateUserLicensesOperation

CancelOperation(context.Context, \*longrunningpb.CancelOperationRequest, ...gax.CallOption) error

GetOperation(context.Context, \*longrunningpb.GetOperationRequest, ...gax.CallOption)

(\*longrunningpb.Operation, error)

ListOperations(context.Context, \*longrunningpb.ListOperationsRequest, ...gax.CallOption) \*OperationIterator

}

// UserLicenseClient is a client for interacting with Discovery Engine API.

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with

```

method calls.
//
// Service for managing
// User Licenses.
type UserLicenseClient struct {
 // The internal transport-dependent client.
 internalClient internalUserLicenseClient

 // The call options for this service.
 CallOptions *UserLicenseCallOptions

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient *lroauto.OperationsClient
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *UserLicenseClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *UserLicenseClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns
// a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *UserLicenseClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// ListUserLicenses lists the User Licenses.
func (c *UserLicenseClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
 return c.internalClient.ListUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicenses updates the User License.

```

```

// This method is used for batch assign/unassign licenses to users.
func (c *UserLicenseClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 return c.internalClient.BatchUpdateUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
//
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *UserLicenseClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 return c.internalClient.BatchUpdateUserLicensesOperation(name)
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest,
opts ...gax.CallOption) error {
 return c.internalClient.CancelOperation(ctx, req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts
...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) ListOperations(ctx context.Context,
req *longrunningpb.ListOperationsRequest, opts ...gax.CallOption) *OperationIterator {
 return c.internalClient.ListOperations(ctx, req, opts...)
}

// userLicenseGRPCClient is a client for interacting with Discovery Engine API over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing UserLicenseClient
 CallOptions **UserLicenseCallOptions

 // The gRPC API client.
 userLicenseClient discoveryenginepb.UserLicenseServiceClient

```

```

// LROClient is used internally to handle long-running operations.
// It is exposed so that its CallOptions can be modified if required.
// Users should not Close this client.
LROClient **lroauto.OperationsClient

operationsClient longrunningpb.OperationsClient

// The x-goog-* metadata
// to be sent with each request.
xGoogHeaders []string

logger *slog.Logger
}

// NewUserLicenseClient creates a new user license service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing User Licenses.
func NewUserLicenseClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := defaultUserLicenseGRPCClientOptions()
 if newUserLicenseClientHook != nil {
 hookOpts, err := newUserLicenseClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }
 client := UserLicenseClient{CallOptions: defaultUserLicenseCallOptions()}

 c := &userLicenseGRPCClient{
 connPool: connPool,
 userLicenseClient: discoveryenginepb.NewUserLicenseServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
 }
 c.setGoogleClientInfo()

 client.internalClient = c

 client.LROClient, err = lroauto.NewOperationsClient(ctx, gtransport.WithConnPool(connPool))
 if err != nil {

```

```

// This error "should not happen", since we are just reusing old connection pool
// and never actually need to dial.
// If this does happen, we could leak connp. However, we cannot close conn:
// If the user invoked the constructor with option.WithGRPCConn,
// we would close a connection that's still in use.
// TODO: investigate error conditions.
return nil, err
}
c.LROClient = &client.LROClient
return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *userLicenseGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc", grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *userLicenseGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // LROClient is used internally

```

```

to handle long-running operations.
// It is exposed so that its CallOptions can be modified if required.
// Users should not Close this client.
LROClient **lroauto.OperationsClient

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing UserLicenseClient
CallOptions **UserLicenseCallOptions

logger *slog.Logger
}

// NewUserLicenseRESTClient creates a new user license service rest client.
//
// Service for managing User Licenses.
func NewUserLicenseRESTClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := append(defaultUserLicenseRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultUserLicenseRESTCallOptions()
 c := &userLicenseRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 lroOpts
 := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
 }
 opClient, err := lroauto.NewOperationsRESTClient(ctx, lroOpts...)
 if err != nil {
 return nil, err
 }
 c.LROClient = &opClient

 return &UserLicenseClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultUserLicenseRESTClientOptions() []option.ClientOption {

```

```

return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://discoveryengine.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://discoveryengine.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://discoveryengine.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
}
}

// setGoogleClientInfo
// sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *userLicenseRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *userLicenseRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *userLicenseGRPCClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest,
opts ...gax.CallOption) *UserLicenseIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).ListUserLicenses[0:len((*c.CallOptions).ListUserLicenses):len((*c.CallOptions).ListUser
Licenses)], opts...)
 it := &UserLicenseIterator{ }

```

```

req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
 resp := &discoveryenginepb.ListUserLicensesResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx,
 c.userLicenseClient.ListUserLicenses, req, settings.GRPC, c.logger, "ListUserLicenses")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *userLicenseGRPCClient) BatchUpdateUserLicenses(ctx context.Context, req
 *discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
 (*BatchUpdateUserLicensesOperation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders,
 hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)

```



```

 opts =
append((*c.CallOptions).BatchUpdateUserLicenses[0:len((*c.CallOptions).BatchUpdateUserLicenses):len((*c.CallOptions).BatchUpdateUserLicenses)], opts...)
 var resp *longrunningpb.Operation
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.userLicenseClient.BatchUpdateUserLicenses, req, settings.GRPC, c.logger,
"BatchUpdateUserLicenses")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 }, nil
}

func (c *userLicenseGRPCClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx,
hds...)
 opts =
append((*c.CallOptions).CancelOperation[0:len((*c.CallOptions).CancelOperation):len((*c.CallOptions).CancelOp
eration)], opts...)
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 _, err = executeRPC(ctx, c.operationsClient.CancelOperation, req, settings.GRPC, c.logger, "CancelOperation")
 return err
 }, opts...)
 return err
}

func (c *userLicenseGRPCClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 var resp *longrunningpb.Operation
 err := gax.Invoke(ctx, func(ctx context.Context,

```

```

settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

func (c *userLicenseGRPCClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,
opts ...gax.CallOption) *OperationIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).ListOperations[0:len((*c.CallOptions).ListOperations):len((*c.CallOptions).ListOperations
)], opts...)
 it := &OperationIterator{ }
 req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
 it.InternalFetch = func(pageSize int, pageToken string) (*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{ }
 if pageToken !=
 "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.ListOperations, req, settings.GRPC, c.logger, "ListOperations")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetOperations(), resp.GetNextPageToken(), nil
 }
 fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {

```

```

 return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// ListUserLicenses lists the
// User Licenses.
func (c *userLicenseRESTClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
 it := &UserLicenseIterator{}
 req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
 resp := &discoveryenginepb.ListUserLicensesResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/userLicenses", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v",
req.GetFilter()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }
 }
}

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListUserLicenses")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response
= resp
return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it

```

```

}

// BatchUpdateUserLicenses updates the User License.
// This method is used for batch assign/unassign licenses to users.
func (c *userLicenseRESTClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
 err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:batchUpdateUserLicenses", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err
:= executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "BatchUpdateUserLicenses")
 if err != nil {
 return err

```

```

 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:cancel", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery
 = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 return gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path

```

```

 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 _, err = executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "CancelOperation")
 return err
}, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption)
(*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET",
baseUrl.String(), nil)
 if err != nil {
 return err
 }

```

```

httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,
opts ...gax.CallOption) *OperationIterator {
 it := &OperationIterator{}
 req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
 unmarshalOptions := &protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{}
 if pageToken != "" {
 req.PageToken =
 pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/operations", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 }
}

```



```

if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
= settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListOperations")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetOperations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

```

```

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// BatchUpdateUserLicensesOperation returns a new
// BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *userLicenseGRPCClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
return &BatchUpdateUserLicensesOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
}
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *userLicenseRESTClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
override := fmt.Sprintf("/v1/%s", name)
return &BatchUpdateUserLicensesOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
pollPath: override,
}
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

```

```

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var
 newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
 Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
 return &LicensesCallOptions{
 Delete: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier:

```

```

1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
}),
},
GetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

```

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.  
type internalLicensesClient interface {

```

Close() error
setGoogleClientInfo(...string)
Connection() *grpc.ClientConn
Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
TestIamPermissions(context.Context,
*computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

```

// LicensesClient is a client for interacting with Google Compute Engine API.

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

//

// The Licenses API.

type LicensesClient struct {

// The internal transport-dependent client.

internalClient internalLicensesClient

// The call options for this service.

CallOptions \*LicensesCallOptions

}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when

// the client is no longer required.

func (c \*LicensesClient) Close() error {

return c.internalClient.Close()

}

// setGoogleClientInfo sets the name and version of the application in

// the `x-goog-api-client` header

passed on each request. Intended for

// use by Google-written clients.

func (c \*LicensesClient) setGoogleClientInfo(keyval ...string) {

c.internalClient.setGoogleClientInfo(keyval...)

}

// Connection returns a connection to the API service.

//

// Deprecated: Connections are now pooled so this method does not always

```

// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context,
 req *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available
// in the specified project. This method does not get any licenses that belong to other projects, including licenses
// attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this
// method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This
// resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
// resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return

```

```

c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type
licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicensesClient
 CallOptions **LicensesCallOptions

 logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

```

```

}

callOpts := defaultLicensesRESTCallOptions()
c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {
 return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo sets the name and version of the application
// in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

```



```

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {
 return err
 }
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for
// use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

if settings.Path != "" {
 baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

```

```

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.License{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

```

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/getIamPolicy",
req.GetProject(), req.GetResource())

 params := url.Values{}

```

```

if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
{
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-

```

party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err
 := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
 err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }
 })
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended

for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
...gax.CallOption) *LicenseIterator {

```

```

 it := &LicenseIterator{}
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
 resp := &computepb.LicensesListResponse{}
 if pageToken != "" {
 req.PageToken = proto.String(pageToken)
 }
 if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
 } else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}

```

```

if req != nil && req.Filter !=
nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil

```

```

}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating
Cloud Marketplace images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}

```



```

resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
TestIamPermissions(ctx context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts
...gax.CallOption) (*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
 req.GetResource())

```

```

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
m := protojson.MarshalOptions{AllowPartial: true}
body := req.GetLicenseResource()
jsonReq, err := m.Marshal(body)

```

```

if err != nil {
 return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

params := url.Values{}
if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}
if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {

```

```

 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project:
req.GetProject(),
 },
}
return op, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

```

```

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GranteeType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GranteeType = 3
)

// Enum value maps for Permission_GranteeType.
var (
 Permission_GranteeType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GranteeType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GranteeType) Enum() *Permission_GranteeType {
 p := new(Permission_GranteeType)
 *p = x
 return p
}

func (x Permission_GranteeType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

```

```

}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

```

```

)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache

```

unknownFields protoimpl.UnknownFields

// Output only. Identifier. The permission name. A unique name will be

// generated on create. Examples:

//

// tunedModels/{tuned\_model}/permissions/{permission}

// corpora/{corpus}/permissions/{permission}

//

// Output only.

Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`

// Optional. Immutable. The type of the grantee.

GranteeType \*Permission\_GranteeType

`protobuf:"varint,2,opt,name=grantee\_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission\_GranteeType,oneof"

json:"grantee\_type,omitempty"`

// Optional. Immutable. The email address of the user of group which this

// permission refers. Field is not set when permission's grantee type is

// EVERYONE.

EmailAddress \*string `protobuf:"bytes,3,opt,name=email\_address,json=emailAddress,proto3,oneof"

json:"email\_address,omitempty"`

// Required. The role granted by this permission.

Role \*Permission\_Role

`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission\_Role,oneof"

json:"role,omitempty"`

}

func (x \*Permission) Reset() {

\*x = Permission{}

mi := &file\_google\_ai\_generativelanguage\_v1beta\_permission\_proto\_msgTypes[0]

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

ms.StoreMessageInfo(mi)

}

func (x \*Permission) String() string {

return protoimpl.X.MessageStringOf(x)

}

func (\*Permission) ProtoMessage() {}

func (x \*Permission) ProtoReflect() protoreflect.Message

{

mi := &file\_google\_ai\_generativelanguage\_v1beta\_permission\_proto\_msgTypes[0]

if x != nil {

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

if ms.LoadMessageInfo() == nil {

ms.StoreMessageInfo(mi)

}

return ms



```

}
return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
 0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,

```

0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,  
0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,  
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,  
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,  
0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,  
0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,  
0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,  
0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,  
0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,  
0x01, 0x01, 0x12,  
0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,  
0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,  
0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,  
0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,  
0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,  
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,  
0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,  
0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,  
0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,  
0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,  
0x54, 0x59,  
0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,  
0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,  
0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,  
0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,  
0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,  
0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,  
0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,  
0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,  
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,  
0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,  
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,  
0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,  
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,  
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,  
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,  
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,  
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,  
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,  
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,  
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,  
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,

```

0x6f, 0x6c, 0x65, 0x42,
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc
)

```

```

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GranteeType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}
var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1beta.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
 google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

```

```

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }
func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
if File_google_ai_generativelanguage_v1beta_permission_proto != nil {
return
}
file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
NumEnums: 2,
NumMessages: 1,
NumExtensions: 0,
NumServices: 0,
},
GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1beta_permission_proto = out.File
file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
= nil
file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Code generated by protoc-gen-go_gapic. DO NOT EDIT.
//go:build go1.23
package discoveryengine_test

```

```

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
)

func ExampleUserLicenseClient_ListUserLicenses_all() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 for resp, err := range c.ListUserLicenses(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

func ExampleUserLicenseClient_ListOperations_all() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```
defer c.Close()
```

```
req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
}
for resp, err := range c.ListOperations(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
```

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions



for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <http://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.

// versions:

```

// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 context "context"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
 protoreflect
 "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Assignment protocol for a license pool.
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The type of assignment protocol.
 //
 // Types that are assignable to AssignmentType:
 //
 // *AssignmentProtocol_ManualAssignmentType_
 // *AssignmentProtocol_AutoAssignmentType_
 AssignmentType
 isAssignmentProtocol_AssignmentType `protobuf_oneof:"assignment_type"`
}

func (x *AssignmentProtocol) Reset() {
 *x = AssignmentProtocol{}
 mi :=

```

```

&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol) ProtoMessage() {}

func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType {
 if m != nil {
 return m.AssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
 return x.ManualAssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
 return x.AutoAssignmentType
 }
}

```

```

}
return nil
}

type isAssignmentProtocol_AssignmentType interface {
isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_struct {
// Allow manual assignments triggered by administrative operations only.
ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
`protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type
AssignmentProtocol_AutoAssignmentType_struct {
// Allow automatic assignments triggered by data plane operations.
AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
`protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
}

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Identifier. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Required. Assignment protocol for the license pool.
LicenseAssignmentProtocol *AssignmentProtocol
`protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
json:"license_assignment_protocol,omitempty"`
// Output only. Licenses count that are available to be assigned.
AvailableLicenseCount int32
`protobuf:"varint,3,opt,name=available_license_count,json=availableLicenseCount,proto3"
json:"available_license_count,omitempty"`
// Output only. Total number of licenses in the pool.
TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total_license_count,json=totalLicenseCount,proto3"
json:"total_license_count,omitempty"`
}

func (x *LicensePool) Reset() {

```

```

*x = LicensePool{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *LicensePool) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensePool) ProtoMessage() {}

func (x *LicensePool) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 if
x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.
func (*LicensePool) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{1}
}

func (x *LicensePool) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

```

```

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a
// license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

func (x *GetLicensePoolRequest) Reset() {
 *x = GetLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if
 ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)

```

```

 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The license pool to update.
 //
 // The license pool's name field is used to identify the license pool to
 // update. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`.
 LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
 json:"license_pool,omitempty"`
 //
 // Required. The list of fields to update.
 UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
 json:"update_mask,omitempty"`
}

func (x *UpdateLicensePoolRequest) Reset() {
 *x = UpdateLicensePoolRequest{}
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UpdateLicensePoolRequest) String() string {

```



```

return protoimpl.X.MessageStringOf(x)
}

func (*UpdateLicensePoolRequest) ProtoMessage() {}

func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor
// instead.
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{3}
}

func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {
 if x != nil {
 return x.LicensePool
 }
 return nil
}

func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

// Request message for
//
// [LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

```

```

// Required. License pool name.
Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
// Required. Username.
// Format: `name@domain.com`.
Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func
(x *AssignRequest) Reset() {
*x = AssignRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{4}
}

func (x *AssignRequest) GetParent()
string {
if x != nil {
return x.Parent
}
}

```

```

return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x
 != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{5}
}

// Request message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset()
{
 *x = UnassignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
}

```

```

return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{6}
}

func (x *UnassignRequest) GetParent() string {
 if x != nil {
 return
x.Parent
 }
 return ""
}

func (x *UnassignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementSer
vice.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

```

```

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. The maximum number of users to return. The service may return
 // fewer than this value.
 PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3"
 json:"page_size,omitempty"`
 // Optional. A page token, received from a previous `EnumerateLicensedUsers`
 // call. Provide this to retrieve the subsequent page.
 PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

func (x *EnumerateLicensedUsersRequest) Reset() {
 *x = EnumerateLicensedUsersRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

```

```

}

func (x *EnumerateLicensedUsersRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo()
 == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{8}
}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

```

```

}

// A licensed user.
type LicensedUser struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Username.
 // Format: `name@domain.com`.
 Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
 //
 // Output only. Timestamp when the license was assigned.
 AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3"
 json:"assign_time,omitempty"`
 // Output only. Timestamp when the license was recently used. This may not be
 // the most recent usage time, and will be updated regularly (within 24
 // hours).
 RecentUsageTime *timestamppb.Timestamp
 `protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3"
 json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {
 *x = LicensedUser{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
}

```



```

return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }
 return nil
}

// Response message for
//
// [LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
// ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The list of licensed users.
 LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
 json:"licensed_users,omitempty"`

 // A token that can be sent as `page_token` to retrieve the next page.
 // If this field is omitted, there are no subsequent pages.
 NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
 json:"next_page_token,omitempty"`
}

```

```

}

func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersResponse) ProtoMessage() {}

func
(x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{10}
}

func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
 return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
}

```

```

return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated:
// Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {

```

```

state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Optional. The time to live for an inactive license. After this time has
// passed, the license will be automatically unassigned from the user. Must
// be at least 7 days, if set. If unset, the license will never expire.
InactiveLicenseTtl *durationpb.Duration
`protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration {
 if x != nil {

```

```

return x.InactiveLicenseTtl
}
return nil
}

```

```

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protoreflect.FileDescriptor

```

```

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{

```

```

0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
0x6f,
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,
0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,
0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
0x62, 0x75, 0x66, 0x2f, 0x64,
0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,
0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,
0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,
0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73,
0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,

```

0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01,  
0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42, 0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74,

0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,  
0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69,  
0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,

0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc3, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74,  
0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x89, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74,  
0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x6f, 0xda, 0x41, 0x18, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4e, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x3e, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,



0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12, 0xdf, 0x01, 0x0a, 0x06,  
0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71,  
0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f,  
0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75,  
0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f, 0x3a, 0x01,  
0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62,  
0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a,  
0x2f, 0x6f, 0x72, 0x64,  
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0xe7, 0x01,  
0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73, 0x65, 0x22, 0x5a, 0xda, 0x41, 0x10,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73,  
0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31, 0x2f, 0x7b,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f,  
0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x75,  
0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e, 0x75, 0x6d,  
0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,  
0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e,  
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74,  
0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31,  
0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22,  
0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4c,

```

0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69,
0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f,
0x6f, 0x72, 0x64,
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c,
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63, 0xca, 0x41,
0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f,
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,
0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,
0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75,
0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72,
0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63,
0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x63, 0x6f,
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2f,
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70, 0x69, 0x76,
0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0x3b,
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa, 0x02, 0x2d,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e,
0x43, 0x6f, 0x6d,
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x50,
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x2d,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43, 0x6f, 0x6d,
0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x5c, 0x50,
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x32,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x43,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,
0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x3a, 0x3a,
0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
= file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f

```

```

unc() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
})
return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
 (*AssignmentProtocol)(nil),
 // 0: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 (*LicensePool)(nil), // 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
 (*GetLicensePoolRequest)(nil), // 2:
google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 (*UpdateLicensePoolRequest)(nil), // 3:
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 (*AssignRequest)(nil), // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
 (*AssignResponse)(nil), // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
 (*UnassignRequest)(nil), // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 (*UnassignResponse)(nil), // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 (*EnumerateLicensedUsersRequest)(nil), // 8:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 (*LicensedUser)(nil),
 // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
 (*EnumerateLicensedUsersResponse)(nil), // 10:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
 (*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 (*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 (*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask
 (*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
 (*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =
[]int32{
 11, // 0:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name
-> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12, // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name -
> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
}

```

```

13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
google.protobuf.FieldMask
14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
google.protobuf.Timestamp
14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name
-> google.protobuf.Timestamp
9, // 7:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensedUser
15, // 8:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
ype_name -> google.protobuf.Duration
2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
> google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
3, // 10:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
google.cloud.commerce.consumer.procurement.v1.AssignRequest
6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type
-> google.cloud.commerce.consumer.procurement.v1.UnassignRequest
8, // 13:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_ty
pe -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
1, // 14:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
1, // 15:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
google.cloud.commerce.consumer.procurement.v1.AssignResponse
7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
google.cloud.commerce.consumer.procurement.v1.UnassignResponse
10, // 18:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_ty
pe
-> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14, // [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}

```

```

func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File:
 protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
 }.Build()
 File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs
 = nil
}

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// LicenseManagementServiceClient is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {

```

```

// Gets the license pool.
GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
// Updates the license pool if one exists for this Order.
UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool,
error)
// Assigns a license to a user.
Assign(ctx context.Context,
in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
// Unassigns a license from a user.
Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
// Enumerates all users assigned a license.
EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption)
(*EnumerateLicensedUsersResponse, error)
}

type licenseManagementServiceClient struct {
cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts
...grpc.CallOption) (*LicensePool, error) {
out := new(LicensePool)
err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool", in, out, opts...)
if err != nil {
return nil, err
}
return out, nil
}

func
(c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest,
opts ...grpc.CallOption) (*LicensePool, error) {
out := new(LicensePool)
err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool", in, out,
opts...)
if err != nil {
return nil, err
}
return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption)

```

```

(*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign", in,
out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

// LicenseManagementServiceServer is the server API for LicenseManagementService service.  
type LicenseManagementServiceServer interface {

// Gets the license pool.

GetLicensePool(context.Context, \*GetLicensePoolRequest) (\*LicensePool, error)

// Updates the license pool if one exists for this Order.

UpdateLicensePool(context.Context, \*UpdateLicensePoolRequest) (\*LicensePool, error)

// Assigns a license to a user.

Assign(context.Context, \*AssignRequest) (\*AssignResponse,
error)

// Unassigns a license from a user.

Unassign(context.Context, \*UnassignRequest) (\*UnassignResponse, error)

// Enumerates all users assigned a license.

EnumerateLicensedUsers(context.Context, \*EnumerateLicensedUsersRequest)

(\*EnumerateLicensedUsersResponse, error)

}

```

// UnimplementedLicenseManagementServiceServer can be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer struct {
}

func (*UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)
(*UnassignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest) (*EnumerateLicensedUsersResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

func RegisterLicenseManagementServiceServer(s *grpc.Server, srv LicenseManagementServiceServer) {
 s.RegisterService(&_LicenseManagementService_serviceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(GetLicensePoolRequest)
 if err := dec(in); err != nil
 {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/GetLicensePool",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
 }

```



```

return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
in := new(UpdateLicensePoolRequest)
if err := dec(in); err != nil {
return nil, err
}
if interceptor == nil {
return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
}
info := &grpc.UnaryServerInfo{
Server: srv,
FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/UpdateLicensePool",
}
handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
in := new(AssignRequest)
if err := dec(in); err != nil {
return nil, err
}
if interceptor == nil {
return srv.(LicenseManagementServiceServer).Assign(ctx, in)
}
info := &grpc.UnaryServerInfo{
Server: srv,
FullMethod: "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/Assign",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv
interface{}, ctx context.Context, dec func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
in := new(UnassignRequest)
if err := dec(in); err != nil {
return nil, err
}

```

```

}
if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
}
info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(EnumerateLicensedUsersRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor
 == nil {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod:
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
 }
 return interceptor(ctx, in, info, handler)
}

var _LicenseManagementService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.commerce.consumer.procurement.v1.LicenseManagementService",
 HandlerType: (*LicenseManagementServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "GetLicensePool",
 Handler: _LicenseManagementService_GetLicensePool_Handler,
 },
 {
 MethodName: "UpdateLicensePool",
 Handler: _LicenseManagementService_UpdateLicensePool_Handler,
 },
 },
}

```

```

{
 MethodName: "Assign",
 Handler: _LicenseManagementService_Assign_Handler,
},
{
 MethodName:
"Unassign",
 Handler: _LicenseManagementService_Unassign_Handler,
},
{
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
},
},
Streams: []grpc.StreamDesc{},
Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

// [START discoveryengine_v1_generated_UserLicenseService_BatchUpdateUserLicenses_sync]

package main

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
)

func main() {
 ctx := context.Background()
 // This snippet has been automatically generated

```

```

and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &discoveryenginepb.BatchUpdateUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesReque
 st.
}
op, err := c.BatchUpdateUserLicenses(ctx, req)
if err != nil {
 // TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

// [END discoveryengine_v1_generated_UserLicenseService_BatchUpdateUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

```

```

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{

```

```

// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
}
resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

```

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template
 only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
}

```

```

c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
}
it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*computepb.LicensesListResponse)
}
}

func
ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#SetIamPolicyLicenseRequest.
 }

```



```

resp, err := c.SetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may
 require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()
}

```

```

req := &computepb.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#UpdateLicenseRequest.
}
op, err := c.Update(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

```

# 1.103 grpc-go 1.72.2

## 1.103.1 Available under license :

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

# 1.104 google-golang-org-grpc 1.72.2

## 1.104.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses



granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.  
Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# 1.105 common 0.64.0

## 1.105.1 Available under license :

ISC

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# 1.106 logr 1.4.3

## 1.106.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership  
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.



"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.107 google-api-go-client 0.236.0

## 1.107.1 Available under license :

```
This is the official list of people who can contribute
(and typically have contributed) code to the repository.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
```

# The submission process automatically checks to make sure  
# that people submitting code are listed in this file (by email address).  
#  
# Names should be added to this file only after verifying that  
# the individual or the individual's organization has agreed to  
# the appropriate Contributor License Agreement, found here:  
#  
# <https://cla.developers.google.com/about/google-individual>  
# <https://cla.developers.google.com/about/google-corporate>  
#  
# The CLA can be filled out on the web:  
#  
# <https://cla.developers.google.com/>  
#  
# When adding J Random Contributor's name to this file,  
# either J's name or J's organization's name should be  
# added to the AUTHORS file, depending on whether the  
#  
individual or corporate CLA was used.

# Names should be added to this file like so:  
# Name <email address>  
#  
# An entry with two email addresses specifies that the  
# first address should be used in the submit logs and  
# that the second address should be recognized as the  
# same person when interacting with Rietveld.

# Please keep the list sorted.

Alain Vongsouvanhalainv <alainv@google.com>  
Andrew Gerrand <adg@golang.org>  
Brad Fitzpatrick <bradfitz@golang.org>  
Eric Koleda <ekoleda+devrel@googlers.com>  
Francesc Campoy <campoy@golang.org>  
Garrick Evans <garrick@google.com>  
Glenn Lewis <gmlewis@google.com>  
Ivan Krasin <krasin@golang.org>  
Jason Hall <jasonhall@google.com>  
Johan Euphrosine <proppy@google.com>  
Kostik Shtoyk <kostik@google.com>  
Kunpei Sakai <namusyaka@gmail.com>  
Matthew Dolan <dolan@lightstep.com>  
Matthew Whisenhunt <matt.whisenhunt@gmail.com>  
Michael McGreevy <mcgreevy@golang.org>  
Nick Craig-Wood <nickcw@gmail.com>  
Robbie Trencheny <me@robbiet.us>  
Ross Light <light@google.com>

Sarah

Adams <shadams@google.com>

Scott Van Woudenberg <scottvw@google.com>

Takashi Matsuo <tmatsuo@google.com>

Copyright (c) 2011 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2013 Joshua Tacoma. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.108 otel 1.36.0

### 1.108.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,



use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.109 cloud.google.com/go/auth 0.16.2

## 1.109.1 Available under license :

```
// Copyright 2024 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
```

```

// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
 }

```

```

}
for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

```

```

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

```

package procurement

```

```

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"

```

```

 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport
 "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"

```

)

```
var newLicenseManagementClientHook clientHook
```

```
// LicenseManagementCallOptions contains the retry settings for each method of LicenseManagementClient.
```

```
type LicenseManagementCallOptions struct {
```

```
 GetLicensePool []gax.CallOption
```

```
 UpdateLicensePool []gax.CallOption
```

```
 Assign []gax.CallOption
```

```
 Unassign []gax.CallOption
```

```
 EnumerateLicensedUsers []gax.CallOption
```

```
 GetOperation []gax.CallOption
```

```
}
```

```
func defaultLicenseManagementGRPCClientOptions() []option.ClientOption {
```

```
 return []option.ClientOption{
```

```
 internaloption.WithDefaultEndpoint("cloudcommerceconsumerprocurement.googleapis.com:443"),
```

```
 internaloption.WithDefaultEndpointTemplate("cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN:443")
```

```
,
```

```
 internaloption.WithDefaultMTLSEndpoint("cloudcommerceconsumerprocurement.mtls.googleapis.com:443"),
```

```
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
```

```
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
```

```
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
```

```
 internaloption.EnableJwtWithScope(),
```

```
 internaloption.EnableNewAuthLibrary(),
```

```
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
```

```
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
```

```
 }
```

```
}
```

```
func
```

```
 defaultLicenseManagementCallOptions() *LicenseManagementCallOptions {
```

```
 return &LicenseManagementCallOptions{
```

```
 GetLicensePool: []gax.CallOption{},
```

```
 UpdateLicensePool: []gax.CallOption{},
```

```
 Assign: []gax.CallOption{},
```

```
 Unassign: []gax.CallOption{},
```

```
 EnumerateLicensedUsers: []gax.CallOption{},
```

```
 GetOperation: []gax.CallOption{},
```

```
 }
```

```
}
```

```
func defaultLicenseManagementRESTCallOptions() *LicenseManagementCallOptions {
```

```
 return
```

```
 &LicenseManagementCallOptions{
```

```
 GetLicensePool: []gax.CallOption{},
```

```
 UpdateLicensePool: []gax.CallOption{},
```

```
 Assign: []gax.CallOption{},
```

```

Unassign: []gax.CallOption{},
EnumerateLicensedUsers: []gax.CallOption{},
GetOperation: []gax.CallOption{},
}
}

// internalLicenseManagementClient is an interface that defines the methods available from Cloud Commerce
Consumer Procurement API.
type internalLicenseManagementClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 GetLicensePool(context.Context, *procurementpb.GetLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 UpdateLicensePool(context.Context, *procurementpb.UpdateLicensePoolRequest, ...gax.CallOption)
(*procurementpb.LicensePool, error)
 Assign(context.Context, *procurementpb.AssignRequest, ...gax.CallOption) (*procurementpb.AssignResponse,
error)
 Unassign(context.Context, *procurementpb.UnassignRequest,
...gax.CallOption) (*procurementpb.UnassignResponse, error)
 EnumerateLicensedUsers(context.Context, *procurementpb.EnumerateLicensedUsersRequest, ...gax.CallOption)
*LicensedUserIterator
 GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
}

// LicenseManagementClient is a client for interacting with Cloud Commerce Consumer Procurement API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// Service for managing licenses.
type LicenseManagementClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseManagementClient

 // The call options for this service.
 CallOptions *LicenseManagementCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseManagementClient)
 Close() error {
 return c.internalClient.Close()
 }
}

```

```

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseManagementClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseManagementClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// GetLicensePool gets the license pool.
func (c *LicenseManagementClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 return c.internalClient.GetLicensePool(ctx, req, opts...)
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *LicenseManagementClient)
 UpdateLicensePool(ctx context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 return c.internalClient.UpdateLicensePool(ctx, req, opts...)
}

// Assign assigns a license to a user.
func (c *LicenseManagementClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 return c.internalClient.Assign(ctx, req, opts...)
}

// Unassign unassigns a license from a user.
func (c *LicenseManagementClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest, opts
...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 return c.internalClient.Unassign(ctx, req, opts...)
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *LicenseManagementClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 return c.internalClient.EnumerateLicensedUsers(ctx,
 req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.

```



```

func (c *LicenseManagementClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// licenseManagementGRPCClient is a client for interacting with Cloud Commerce Consumer Procurement API
// over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 // The gRPC API client.
 licenseManagementClient procurementpb.LicenseManagementServiceClient

 operationsClient longrunningpb.OperationsClient

 //
 // The x-goog-* metadata to be sent with each request.
 xGoogHeaders []string

 logger *slog.Logger
}

// NewLicenseManagementClient creates a new license management service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing licenses.
func NewLicenseManagementClient(ctx context.Context, opts ...option.ClientOption) (*LicenseManagementClient,
error) {
 clientOpts := defaultLicenseManagementGRPCClientOptions()
 if newLicenseManagementClientHook != nil {
 hookOpts, err := newLicenseManagementClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
 }

 connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
 if err != nil {
 return nil, err
 }

```

```

client := LicenseManagementClient{CallOptions: defaultLicenseManagementCallOptions()}

c := &licenseManagementGRPCClient{
 connPool:
 connPool,
 licenseManagementClient: procurementpb.NewLicenseManagementServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
}
c.setGoogleClientInfo()

client.internalClient = c

return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *licenseManagementGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc",
 grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

```

```

// The http client.
httpClient *http.Client

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing LicenseManagementClient
CallOptions **LicenseManagementCallOptions

logger *slog.Logger
}

// NewLicenseManagementRESTClient creates a new license management service rest client.
//
// Service for managing licenses.
func NewLicenseManagementRESTClient(ctx context.Context,
 opts ...option.ClientOption) (*LicenseManagementClient, error) {
 clientOpts := append(defaultLicenseManagementRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseManagementRESTCallOptions()
 c := &licenseManagementRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseManagementClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseManagementRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://cloudcommerceconsumerprocurement.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://cloudcommerceconsumerprocurement.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

```

```

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient
 = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseManagementRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *licenseManagementGRPCClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.GetLicensePool, req, settings.GRPC, c.logger,
"GetLicensePool")
 return err
 },
 opts...)
 if err != nil {
 return nil, err
 }
}

```

```

return resp, nil
}

func (c *licenseManagementGRPCClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.UpdateLicensePool, req, settings.GRPC, c.logger,
"UpdateLicensePool")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient)
Assign(ctx context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption)
(*procurementpb.AssignResponse, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 var resp *procurementpb.AssignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Assign, req, settings.GRPC, c.logger, "Assign")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {

```

```

hds := []string{"x-goog-request-params",
fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
var resp *procurementpb.UnassignResponse
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.Unassign, req, settings.GRPC, c.logger, "Unassign")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

```

```

func (c *licenseManagementGRPCClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts
=
append((*c.CallOptions).EnumerateLicensedUsers[0:len((*c.CallOptions).EnumerateLicensedUsers):len((*c.CallOptions).EnumerateLicensedUsers)], opts...)
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.EnumerateLicensedUsers, req, settings.GRPC, c.logger,
"EnumerateLicensedUsers")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }
}

```

```

 }

 it.Response = resp
 return resp.GetLicensedUsers(), resp.GetNextPageToken(),
 nil
 }
 fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
 }

 it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
 it.pageInfo.MaxSize = int(req.GetPageSize())
 it.pageInfo.Token = req.GetPageToken()

 return it
}

func (c *licenseManagementGRPCClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
 opts...)
 var resp *longrunningpb.Operation
 err := gax.Invoke(ctx,
 func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

// GetLicensePool gets the license pool.
func (c *licenseManagementRESTClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 baseUrl, err := url.Parse(c.endpoint)

```

```

if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx,
hds...)
opts =
append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil

```



```

}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *licenseManagementRESTClient) UpdateLicensePool(ctx
context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetLicensePool()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetLicensePool().GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 params.Add("updateMask", string(field[1:len(field)-1]))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds,
"Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &procurementpb.LicensePool{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 })

```

```

httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateLicensePool")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Assign assigns a license to a user.
func
(c *licenseManagementRESTClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:assign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

```

```

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial:
true, DiscardUnknown: true}
resp := &procurementpb.AssignResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Assign")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Unassign unassigns a license from a user.
func (c *licenseManagementRESTClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err :=
 m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }

```

```

}
baseUrl.Path += fmt.Sprintf("/v1/%v:unassign", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.UnassignResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Unassign")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *licenseManagementRESTClient) EnumerateLicensedUsers(ctx context.Context, req

```

```

*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if
pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:enumerateLicensedUsers", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
= settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "EnumerateLicensedUsers")
 if err != nil {

```

```

 return err
}
if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetOperation is a utility method from
google.longrunning.Operations.
func (c *licenseManagementRESTClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

```

// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var
 newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
 Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
 return &LicensesCallOptions{
 Delete: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 },

```



```

Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier:
1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
GetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{

```

```

 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 TestIamPermissions(context.Context,
 *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
 Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.
 CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed
// on each request. Intended for
// use by Google-written clients.

```

```

func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context, req
*computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available in the
// specified project. This method does not get any licenses that belong to other projects, including licenses attached to
// publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to
// make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

```

```

}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type
licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicensesClient
 CallOptions **LicensesCallOptions

 logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.

```

```
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()
 c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
 }
 opC, err := NewGlobalOperationsRESTClient(ctx, o...)
 if err != nil {
 return nil, err
 }
 c.operationClient = opC

 return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo sets the name and version of the application in
//
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
```

```

func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {
 return err
 }
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for use
// only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)

```

```

hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {

```

```

 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.License{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
{
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)

```



```

if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy",
req.GetProject(), req.GetResource())

params := url.Values{}
if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if
settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)

```

```

 if e != nil {
 return nil, e
 }
 return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
 err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err
 := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err

```

```

 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
...gax.CallOption) *LicenseIterator {
 it := &LicenseIterator{}
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
 resp := &computepb.LicensesListResponse{}
 if pageToken != "" {
 req.PageToken = proto.String(pageToken)
 }
 if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
 } else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
 }
 }
}

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

params := url.Values{}
if req != nil && req.Filter != nil {
 params.Add("filter",
fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header
= headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }
}

```

```

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace
images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

```

```

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) TestIamPermissions(ctx
context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }
}

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

```

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Update(ctx context.Context,
req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
m := protojson.MarshalOptions{AllowPartial: true}
body := req.GetLicenseResource()
jsonReq, err := m.Marshal(body)
if err != nil {
return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

params := url.Values{}
if req != nil && req.RequestId != nil {
params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}
if req != nil && req.UpdateMask != nil {
params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

```



```

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return
op, nil
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"

```

```

longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
"google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := procurement.NewLicenseManagementRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseManagementClient_Assign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
}
resp, err := c.Assign(ctx, req)
if err != nil {
 // TODO: Handle error.
}
//
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicensedUsersRequest.
 }
 it := c.EnumerateLicensedUsers(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 //

```

TODO: Use resp.

```
_ = resp

// If you need to access the underlying RPC response,
// you can do so by casting the `Response` as below.
// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
}
}
```

```
func ExampleLicenseManagementClient_GetLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //
 // See
 // https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRequest.
 }
 resp, err := c.GetLicensePool(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}
```

```
func ExampleLicenseManagementClient_Unassign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
```

```

// TODO: Handle error.
}
defer c.Close()

req := &procurementpb.UnassignRequest{
// TODO: Fill request struct fields.
// See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest.
}
resp,
err := c.Unassign(ctx, req)
if err != nil {
// TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_UpdateLicensePool() {
ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
// TODO: Handle error.
}
defer c.Close()

req := &procurementpb.UpdateLicensePoolRequest{
// TODO: Fill request struct fields.
// See
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
}
resp, err := c.UpdateLicensePool(ctx, req)
if err != nil {
// TODO: Handle error.
}
//
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_GetOperation() {
ctx := context.Background()

```

```
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
}
resp, err := c.GetOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without



modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

//go:build go1.23

package compute\_test

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

```
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
```

```
// Code generated by protoc-gen-go_gapic. DO NOT EDIT.
```

```
package compute
```

```
import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
)
```

```
var
 newLicenseCodesClientHook clientHook
```

```
// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}
```

```
func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 }
```

```

 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
}
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 Close()
 error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return c.internalClient.Close()
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.

```

```

//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions

 logger *slog.Logger
}

// NewLicenseCodesRESTClient
// creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)

```

```

httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
if err != nil {
 return nil, err
}

callOpts := defaultLicenseCodesRESTCallOptions()
c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return

```

```

 nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.LicenseCode{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }
 })

```



```

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a
// caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v/testIamPermissions",
 req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds =
 append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
 tIamPermissions)], opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {

```

```

 baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"

```

```

"net/http"
"net/url"
"time"

computePb "cloud.google.com/go/compute/apiv1beta/computePb"
gax "github.com/googleapis/gax-go/v2"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
httpTransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
)

var
newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 }
}

```

```

 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
(*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return
 c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
Marketplace images.
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,

```

```

 req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions

 logger *slog.Logger
}

// NewLicenseCodesRESTClient
// creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

```

```

return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
}
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gopic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
return
nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts

```

```

...gax.CallOption)(*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v", req.GetProject(),
 req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.LicenseCode{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

```

```

// TestIamPermissions returns permissions that

```

a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v/testIamPermissions",
req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds
 = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }
 })
}
```



```

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1alpha/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

```

```

// Defines types of the grantee of this permission.
type Permission_GrantType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {

```

```

return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[int]) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
// The default value. This value is unused.
Permission_ROLE_UNSPECIFIED Permission_Role = 0
// Owner can use, update, share and delete the resource.
Permission_OWNER Permission_Role = 1
// Writer can use, update and share the resource.
Permission_WRITER Permission_Role = 2
// Reader can use the resource.
Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
Permission_Role_name = map[int32]string{
0: "ROLE_UNSPECIFIED",
1: "OWNER",
2: "WRITER",
3: "READER",
}
Permission_Role_value = map[string]int32{
"ROLE_UNSPECIFIED": 0,
"OWNER": 1,
"WRITER": 2,
"READER": 3,
}
)

func (x Permission_Role) Enum() *Permission_Role {
p := new(Permission_Role)
*p = x
return p
}

```

```

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service
// accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}

```

```

//
// Output only.
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Optional. Immutable. The type of the grantee.
GranteeType *Permission_GranteeType
`protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
json:"grantee_type,omitempty"`
// Optional. Immutable. The email address of the user of group which this
// permission refers. Field is not set when permission's grantee type is
// EVERYONE.
EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
json:"role,omitempty"`
}

func (x *Permission) Reset() {
*x = Permission{}
mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

```

```

}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1alpha_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,
 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,
 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,
 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,
 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,
 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,
 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,

```

0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,  
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,  
0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,  
0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,  
0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,  
0x65, 0x54, 0x79, 0x70, 0x65, 0x88,  
0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,  
0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,  
0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,  
0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,  
0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,  
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,  
0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,  
0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,  
0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,  
0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,  
0x4e,  
0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,  
0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,  
0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,  
0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,  
0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,  
0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,  
0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,  
0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,  
0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,  
0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,  
0x2e, 0x63, 0x6f, 0x6d,  
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,  
0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,  
0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,  
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,  
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,  
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,  
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,  
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,  
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,  
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,  
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,  
0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,  
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,  
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,  
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,  
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,  
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,  
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,

```

0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes
= []any{
 (Permission_GrantType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
}
var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}
func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
 if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
 return
 }
 file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
}

```



```

type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
 DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
 EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
 MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
= nil
file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {

```

```

ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
//
It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleLicensesClient_Delete() {
ctx := context.Background()
// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#DeleteLicenseRequest.
}
op, err := c.Delete(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
ctx := context.Background()

```

```

// This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseRequest.
}
resp,
err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetIamPolicyLicenseRequest.
 }
 resp, err := c.GetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

```

```

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded
 // as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#ListLicensesRequest.
 }

```

```

it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _
 = it.Response.(*compute.LicensesListResponse)
}
}

func ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute.LicensesClient#SetIamPolicyLicenseRequest.
 }
 resp, err := c.SetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()

```

```
//
This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := compute.NewLicensesRESTClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseRequest.
}
resp, err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#UpdateLicenseRequest.
 }
 op, err := c.Update(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
}
```

```

if err != nil {
 // TODO: Handle error.
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#ListLicensesRequest.
 }

```

```

}
for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```



```

// TODO: Use client.
_ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request
 struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseCodeRequest.
 }
 resp,

```

```

err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
)

func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()
 // This snippet has been automatically
 // generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicense
 dUsersRequest.
}
for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}

```

```

// Copyright 2023 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

```

```

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

```

package compute

```

```

import (
 "bytes"

```

```

"context"
"fmt"
"log/slog"
"math"
"net/http"
"net/url"
"time"

computepb "cloud.google.com/go/compute/apiv1beta/computepb"
gax "github.com/googleapis/gax-go/v2"
"google.golang.org/api/iterator"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
"google.golang.org/protobuf/proto"
)

var
newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
Delete []gax.CallOption
Get []gax.CallOption
GetIamPolicy []gax.CallOption
Insert []gax.CallOption
List []gax.CallOption
SetIamPolicy []gax.CallOption
TestIamPermissions []gax.CallOption
Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
return &LicensesCallOptions{
Delete: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier:
1.30,
},

```

```

 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
GetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 }),
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

```

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.

```
type internalLicensesClient interface {
```

```
 Close() error
```

```
 setGoogleClientInfo(...string)
```

```

Connection() *grpc.ClientConn
Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
error)
TestIamPermissions(context.Context,
*computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

```

// LicensesClient is a client for interacting with Google Compute Engine API.

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

//

// The Licenses API.

```
type LicensesClient struct {
```

// The internal transport-dependent client.

```
internalClient internalLicensesClient
```

// The call options for this service.

```
CallOptions *LicensesCallOptions
```

```
}
```

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when

// the client is no longer required.

```
func (c *LicensesClient) Close() error {
```

```
 return c.internalClient.Close()
```

```
}
```

// setGoogleClientInfo sets the name and version of the application in

// the `x-goog-api-client` header

passed on each request. Intended for

// use by Google-written clients.

```
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
```

```
 c.internalClient.setGoogleClientInfo(keyval...)
}
```

// Connection returns a connection to the API service.

//

// Deprecated: Connections are now pooled so this method does not always

// return the same resource.

```
func (c *LicensesClient) Connection() *grpc.ClientConn {
```

```

return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context,
 req *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available
// in the specified project. This method does not get any licenses that belong to other projects, including licenses
// attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this
// method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This
// resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
// resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}

```

```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type
licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.
 operationClient *GlobalOperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicensesClient
 CallOptions **LicensesCallOptions

 logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

```



```

callOpts := defaultLicensesRESTCallOptions()
c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opC, err := NewGlobalOperationsRESTClient(ctx, o...)
if err != nil {
 return nil, err
}
c.operationClient = opC

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo sets the name and version of the application
// in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when

```

```

// the client is no longer required.
func (c *licensesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 if err := c.operationClient.Close(); err != nil {
 return err
 }
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Delete deletes the specified license. Caution This resource is intended for
// use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path

```

```

 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)

```

```

hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.License{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.  
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/getIamPolicy",
req.GetProject(), req.GetResource())

```

```

params := url.Values{}
if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
}

```

```

}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
{
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

func (c \*licensesRESTClient) Insert(ctx context.Context, req \*computepb.InsertLicenseRequest, opts

```

...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err
 := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &compute.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
 err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }
 })
}

```

```

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended

for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
```

```
...gax.CallOption) *LicenseIterator {
```

```
 it := &LicenseIterator{}
```

```
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
```

```
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
```

```
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
```

```
 resp := &computepb.LicensesListResponse{}
```

```
 if pageToken != "" {
```

```
 req.PageToken = proto.String(pageToken)
```

```
 }
```

```
 if pageSize > math.MaxInt32 {
```

```
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
```

```
 } else if pageSize != 0 {
```

```
 req.MaxResults = proto.Uint32(uint32(pageSize))
```

```
 }
```

```
 baseUrl, err := url.Parse(c.endpoint)
```

```
 if err != nil {
```

```
 return nil, "", err
```

```
 }
```

```
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())
```

```
 params := url.Values{}
```

```
 if req != nil && req.Filter !=
```

```

nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
}
if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}
if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
}
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

```



```

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating
Cloud Marketplace images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}

```

```

e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
TestIamPermissions(ctx context.Context, req *compute.TestIamPermissionsLicenseRequest, opts
...gax.CallOption) (*compute.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
 req.GetResource())

 // Build HTTP headers from client and context metadata.

```

```

hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.TestPermissionsResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
m := protojson.MarshalOptions{AllowPartial: true}
body := req.GetLicenseResource()
jsonReq, err := m.Marshal(body)
if err != nil {

```

```

return nil, err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

params := url.Values{}
if req != nil && req.RequestId != nil {
params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}
if req != nil && req.UpdateMask != nil {
params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project:
 req.GetProject(),
 },
}
return op, nil
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

```

```

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GrantType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GrantType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GrantType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GrantType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protorelect.EnumNumber(x))
}

```

```

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
 []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

```

```

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

```



```

// Output only. Identifier. The permission name. A unique name will be
// generated on create. Examples:
//
// tunedModels/{tuned_model}/permissions/{permission}
// corpora/{corpus}/permissions/{permission}
//
// Output only.
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Optional. Immutable. The type of the grantee.
GranteeType *Permission_GranteeType
`protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission_GranteeType,oneof"
json:"grantee_type,omitempty"`
// Optional. Immutable. The email address of the user of group which this
// permission refers. Field is not set when permission's grantee type is
// EVERYONE.
EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission_Role,oneof"
json:"role,omitempty"`
}

func (x *Permission) Reset() {
*x = Permission{}
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
}

```

```

return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
 File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
 0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,
 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,

```

0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,  
0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,  
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,  
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,  
0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,  
0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,  
0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,  
0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,  
0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,  
0x01, 0x01, 0x12,  
0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,  
0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,  
0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,  
0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,  
0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,  
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,  
0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,  
0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,  
0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,  
0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,  
0x54, 0x59,  
0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,  
0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,  
0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,  
0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,  
0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,  
0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,  
0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,  
0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,  
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,  
0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,  
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,  
0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,  
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,  
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,  
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,  
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,  
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,  
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,  
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,  
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,  
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,  
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,  
0x6f, 0x6c, 0x65, 0x42,  
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,

```

0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc
)

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GrantType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}

var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1beta.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
 google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }

```

```

func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
if File_google_ai_generativelanguage_v1beta_permission_proto != nil {
return
}
file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
NumEnums: 2,
NumMessages: 1,
NumExtensions: 0,
NumServices: 0,
},
GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1beta_permission_proto = out.File
file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
= nil
file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,



indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

```

// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 context "context"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
 protoreflect
 "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Assignment protocol for a license pool.
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

```

```

// The type of assignment protocol.
//
// Types that are assignable to AssignmentType:
//
// *AssignmentProtocol_ManualAssignmentType_
// *AssignmentProtocol_AutoAssignmentType_
AssignmentType
isAssignmentProtocol_AssignmentType `protobuf_oneof:"assignment_type"`
}

func (x *AssignmentProtocol) Reset() {
 *x = AssignmentProtocol{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol) ProtoMessage() {}

func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType {
 if m != nil {
 return m.AssignmentType
 }
}

```

```

return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {
if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
return x.ManualAssignmentType
}
return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
return x.AutoAssignmentType
}
return nil
}

type isAssignmentProtocol_AssignmentType interface {
isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_ struct {
// Allow manual assignments triggered by administrative operations only.
ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
`protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type
AssignmentProtocol_AutoAssignmentType_ struct {
// Allow automatic assignments triggered by data plane operations.
AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
`protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
}

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {
state protoimpl.MessageState
sizeCache protoimpl.SizeCache
unknownFields protoimpl.UnknownFields

// Identifier. Format:
// `billingAccounts/{billing_account}/orders/{order}/licensePool`
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Required. Assignment protocol for the license pool.

```

```

LicenseAssignmentProtocol *AssignmentProtocol
`protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
json:"license_assignment_protocol,omitempty"`
// Output only. Licenses count that are available to be assigned.
AvailableLicenseCount int32
`protobuf:"varint,3,opt,name=available_license_count,json=availableLicenseCount,proto3"
json:"available_license_count,omitempty"`
// Output only. Total number of licenses in the pool.
TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total_license_count,json=totalLicenseCount,proto3"
json:"total_license_count,omitempty"`
}

func (x *LicensePool) Reset() {
 *x = LicensePool{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensePool) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensePool) ProtoMessage() {}

func (x *LicensePool) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[1]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.
func (*LicensePool) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{1}
}

func (x *LicensePool) GetName() string {

```

```

if x != nil {
 return x.Name
}
return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a
// license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

func (x *GetLicensePoolRequest) Reset() {
 *x = GetLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {

```

```

return protoimpl.X.MessageStringOf(x)
}

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if
 ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The license pool to update.
 //
 // The license pool's name field is used to identify the license pool to
 // update. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`.
 LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
 json:"license_pool,omitempty"`
}

```

Required. The list of fields to update.

```
UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
json:"update_mask,omitempty"`
}
```

```
func (x *UpdateLicensePoolRequest) Reset() {
 *x = UpdateLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}
```

```
func (x *UpdateLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}
```

```
func (*UpdateLicensePoolRequest) ProtoMessage() {}
```

```
func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}
```

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor instead.

```
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{3}
}
```

```
func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {
 if x != nil {
 return x.LicensePool
 }
 return nil
}
```

```
func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {
```



```

if x != nil {
 return x.UpdateMask
}
return nil
}

// Request message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func
(x *AssignRequest) Reset() {
 *x = AssignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{4}
}

func (x *AssignRequest) GetParent()
string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

```

```

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x
 != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{5}
}

// Request message for
//
// [LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset()
{
 *x = UnassignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{6}
}

func (x *UnassignRequest) GetParent() string {
 if x != nil {
 return
 x.Parent
 }
 return ""
}

func (x *UnassignRequest) GetUsername() []string {
 if x != nil {
 return x.Username
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

```

```

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//
// [LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
// ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. The maximum number of users to return. The service may return

```

```

// fewer than this value.
PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3"
json:"page_size,omitempty"`
// Optional. A page token, received from a previous `EnumerateLicensedUsers`
// call. Provide this to retrieve the subsequent page.
PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

func (x *EnumerateLicensedUsersRequest) Reset() {
*x = EnumerateLicensedUsersRequest{}
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo()
== nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {
return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{8}
}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
if x != nil {
return x.Parent
}
return ""
}

```

```

}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

// A licensed user.
type LicensedUser struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Username.
 // Format: `name@domain.com`.
 Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
 //
 // Output only. Timestamp when the license was assigned.
 AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3" json:"assign_time,omitempty"`
 // Output only. Timestamp when the license was recently used. This may not be
 // the most recent usage time, and will be updated regularly (within 24
 // hours).
 RecentUsageTime *timestamppb.Timestamp
 `protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3" json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {
 *x = LicensedUser{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }
 return nil
}

// Response message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License

```



```

ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The list of licensed users.
 LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
 json:"licensed_users,omitempty"`
 // A token that can be sent as `page_token` to retrieve the next page.
 // If this field is omitted, there are no subsequent pages.
 NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
 json:"next_page_token,omitempty"`
}

func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersResponse) ProtoMessage() {}

func
(x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),

```

```

[]int{10}
}

func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
 return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType
struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 }
 return ms
}

```

```

}
return mi.MessageOf(x)
}

// Deprecated:
// Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Optional. The time to live for an inactive license. After this time has
 // passed, the license will be automatically unassigned from the user. Must
 // be at least 7 days, if set. If unset, the license will never expire.
 InactiveLicenseTtl *durationpb.Duration
 `protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
 json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 }
}

```

```

 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration {
 if x != nil {
 return x.InactiveLicenseTtl
 }
 return nil
}

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protoreflect.FileDescriptor

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{
 0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
 0x6f,
 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
 0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
 0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,
 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,
 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x64,
 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,
 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,

```

0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,  
0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73,  
0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,  
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,  
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73,  
0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01,  
0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42, 0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,

0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52,  
0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02,

0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,  
0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69,  
0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,  
0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc3, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74,  
0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x89, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,

0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74,  
0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x6f, 0xda, 0x41, 0x18, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4e, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x3e, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,  
0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12, 0xdf, 0x01, 0x0a, 0x06,  
0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71,  
0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f,  
0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75,  
0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f, 0x3a, 0x01,  
0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62,  
0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a,  
0x2f, 0x6f, 0x72, 0x64,  
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0xe7, 0x01,  
0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73, 0x65, 0x22, 0x5a, 0xda, 0x41, 0x10,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73,  
0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31, 0x2f, 0x7b,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f,  
0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x75,



0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e, 0x75, 0x6d,  
0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,  
0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e,  
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74,  
0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31,  
0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22,  
0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4c,  
0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69,  
0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f,  
0x6f, 0x72, 0x64,  
0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63, 0xca, 0x41,  
0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,  
0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75,  
0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72,  
0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,  
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63,  
0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2f,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70, 0x69, 0x76,  
0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0x3b,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa, 0x02, 0x2d,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x43, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x50,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x2d,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43, 0x6f, 0x6d,  
0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x5c, 0x50,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x32,  
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x43,  
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,  
0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x3a, 0x3a,

```

0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
= file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f
unc() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
 })
 return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
 (*AssignmentProtocol)(nil),
 // 0: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 (*LicensePool)(nil), // 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
 (*GetLicensePoolRequest)(nil), // 2:
 google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 (*UpdateLicensePoolRequest)(nil), // 3:
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 (*AssignRequest)(nil), // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
 (*AssignResponse)(nil), // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
 (*UnassignRequest)(nil), // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 (*UnassignResponse)(nil), // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 (*EnumerateLicensedUsersRequest)(nil), // 8:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 (*LicensedUser)(nil),
 // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
 (*EnumerateLicensedUsersResponse)(nil), // 10:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
 (*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 (*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 (*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask

```

```

(*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
(*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =
[]int32{
 11, // 0:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name
-> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12, // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name -
> google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
 13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
google.protobuf.FieldMask
 14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
google.protobuf.Timestamp
 14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name
-> google.protobuf.Timestamp
 9, // 7:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name ->
google.cloud.commerce.consumer.procurement.v1.LicensedUser
 15, // 8:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
ype_name -> google.protobuf.Duration
 2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
> google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 3, // 10:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
google.cloud.commerce.consumer.procurement.v1.AssignRequest
 6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type
-> google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 8, // 13:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_typ
e -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 1, // 14:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
 1, // 15:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
google.cloud.commerce.consumer.procurement.v1.LicensePool
 5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
google.cloud.commerce.consumer.procurement.v1.AssignResponse
 7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
google.cloud.commerce.consumer.procurement.v1.UnassignResponse

```

```

10, // 18:
google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_type
pe
-> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14, // [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}
func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File:
 protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
 }.Build()
 File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs
 = nil
}

```

```

// Reference imports to suppress errors if they are not otherwise used.
var _ context.Context
var _ grpc.ClientConnInterface

// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
const _ = grpc.SupportPackageIsVersion6

// LicenseManagementServiceClient is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {
 // Gets the license pool.
 GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error)
}

type licenseManagementServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
 return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool", in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

func
(c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest,
opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/UpdateLicensePool", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption)
(*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/Assign", in,
out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, "/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/Unassign",
in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)
 err := c.cc.Invoke(ctx,
"/google.cloud.commerce.consumer.prourement.v1.LicenseManagementService/EnumerateLicensedUsers", in, out,
opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

// LicenseManagementServiceServer is the server API for LicenseManagementService service.

```

type LicenseManagementServiceServer interface {
 // Gets the license pool.
 GetLicensePool(context.Context, *GetLicensePoolRequest) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(context.Context, *UpdateLicensePoolRequest) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(context.Context, *AssignRequest) (*AssignResponse,
 error)
 // Unassigns a license from a user.
 Unassign(context.Context, *UnassignRequest) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(context.Context, *EnumerateLicensedUsersRequest)
 (*EnumerateLicensedUsersResponse, error)
}

// UnimplementedLicenseManagementServiceServer can be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer struct {
}

func (*UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)
(*UnassignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}
func (*UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest) (*EnumerateLicensedUsersResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

func RegisterLicenseManagementServiceServer(s *grpc.Server, srv LicenseManagementServiceServer) {
 s.RegisterService(&_LicenseManagementService_serviceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(GetLicensePoolRequest)

```

```

if err := dec(in); err != nil
{
 return nil, err
}
if interceptor == nil {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, in)
}
info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(UpdateLicensePoolRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool",
 }
 handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(AssignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Assign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,

```



```

FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign",
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv
interface{}, ctx context.Context, dec func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
 in := new(UnassignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(EnumerateLicensedUsersRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor
 == nil {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod:
"/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers",
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
 }
 return interceptor(ctx, in, info, handler)
}

```

```

}

var _LicenseManagementService_serviceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.commerce.consumer.procurement.v1.LicenseManagementService",
 HandlerType: (*LicenseManagementServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "GetLicensePool",
 Handler: _LicenseManagementService_GetLicensePool_Handler,
 },
 {
 MethodName: "UpdateLicensePool",
 Handler: _LicenseManagementService_UpdateLicensePool_Handler,
 },
 {
 MethodName: "Assign",
 Handler: _LicenseManagementService_Assign_Handler,
 },
 {
 MethodName: "Unassign",
 Handler: _LicenseManagementService_Unassign_Handler,
 },
 {
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

```

```

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO:
 // Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute_pb#GetLicenseCodeRequest.
 }
 resp, err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

// TODO: Use resp.
_ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseCodeRequest.
 }
 resp,
 err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

```

```

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.

```

```

// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
}
resp, err := c.GetIamPolicy(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

```

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

 req := &compute.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

```

```

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template
 only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#ListLicensesRequest.
}
it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*computepb.LicensesListResponse)
}
}

func
ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#SetIamPolicyLicenseRequest.
 }
 resp, err := c.SetIamPolicy(ctx, req)

```



```

if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may
 require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.UpdateLicenseRequest{

```

```

// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#UpdateLicenseRequest.
}
op, err := c.Update(ctx, req)
if err != nil {
// TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
// TODO: Handle error.
}
}

```

# 1.110 google-golang-org-api 0.236.0

## 1.110.1 Available under license :

```

This is the official list of people who can contribute
(and typically have contributed) code to the repository.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
The submission process automatically checks to make sure
that people submitting code are listed in this file (by email address).
#
Names should be added to this file only after verifying that
the individual or the individual's organization has agreed to
the appropriate Contributor License Agreement, found here:
#
https://cla.developers.google.com/about/google-individual
https://cla.developers.google.com/about/google-corporate
#
The CLA can be filled out on the web:
#
https://cla.developers.google.com/
#
When adding J Random Contributor's name to this file,
either J's name or J's organization's name should be
added to the AUTHORS file, depending on whether the
#
individual or corporate CLA was used.

Names should be added to this file like so:
Name <email address>
#
An entry with two email addresses specifies that the

```

# first address should be used in the submit logs and  
# that the second address should be recognized as the  
# same person when interacting with Rietveld.

# Please keep the list sorted.

Alain Vongsouvanhalainv <alainv@google.com>  
Andrew Gerrand <adg@golang.org>  
Brad Fitzpatrick <bradfitz@golang.org>  
Eric Koleda <ekoleda+devrel@google.com>  
Francesc Campoy <campoy@golang.org>  
Garrick Evans <garrick@google.com>  
Glenn Lewis <gmlewis@google.com>  
Ivan Krasin <krasin@golang.org>  
Jason Hall <jasonhall@google.com>  
Johan Euphrosine <proppy@google.com>  
Kostik Shtoyk <kostik@google.com>  
Kunpei Sakai <namusyaka@gmail.com>  
Matthew Dolan <dolan@lightstep.com>  
Matthew Whisenhunt <matt.whisenhunt@gmail.com>  
Michael McGreevy <mcgreevy@golang.org>  
Nick Craig-Wood <nickcw@gmail.com>  
Robbie Trencheny <me@robbiet.us>  
Ross Light <light@google.com>  
Sarah  
Adams <shadams@google.com>  
Scott Van Woudenberg <scottvw@google.com>  
Takashi Matsuo <tmatsuo@google.com>  
Copyright (c) 2011 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

- \* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following disclaimer  
in the documentation and/or other materials provided with the  
distribution.
- \* Neither the name of Google Inc. nor the names of its  
contributors may be used to endorse or promote products derived from  
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR  
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2013 Joshua Tacoma. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.111

[google.golang.org/genproto/googleapis/rpc](https://google.golang.org/genproto/googleapis/rpc)  
**0.0.0-20250603155806-513f23925822**

## 1.111.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include



the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.112 go.opentelemetry.io/otel/trace 1.36.0

## 1.112.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.113 google-cloud-go-auth 0.16.2

## 1.113.1 Available under license :

```
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

// [START discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]

package main

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 "google.golang.org/api/iterator"
)

func main() {
 ctx := context.Background()
 // This snippet
 has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 it := c.ListUserLicenses(ctx, req)
```

```

for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove
 this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
}
}

// [END discoveryengine_v1_generated_UserLicenseService_ListUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"

```

```

procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
gax "github.com/googleapis/gax-go/v2"
"google.golang.org/api/iterator"
"google.golang.org/api/option"
"google.golang.org/api/option/internaloption"
gtransport
"google.golang.org/api/transport/grpc"
httptransport "google.golang.org/api/transport/http"
"google.golang.org/grpc"
"google.golang.org/protobuf/encoding/protojson"
"google.golang.org/protobuf/proto"
)

var newLicenseManagementClientHook clientHook

// LicenseManagementCallOptions contains the retry settings for each method of LicenseManagementClient.
type LicenseManagementCallOptions struct {
 GetLicensePool []gax.CallOption
 UpdateLicensePool []gax.CallOption
 Assign []gax.CallOption
 Unassign []gax.CallOption
 EnumerateLicensedUsers []gax.CallOption
 GetOperation []gax.CallOption
}

func defaultLicenseManagementGRPCClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("cloudcommerceconsumerprocurement.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN:443"),
 ,
 internaloption.WithDefaultMTLSEndpoint("cloudcommerceconsumerprocurement.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
 }
}

func
defaultLicenseManagementCallOptions() *LicenseManagementCallOptions {
 return &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 }
}

```



```

 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
}
}

func defaultLicenseManagementRESTCallOptions() *LicenseManagementCallOptions {
 return
 &LicenseManagementCallOptions{
 GetLicensePool: []gax.CallOption{},
 UpdateLicensePool: []gax.CallOption{},
 Assign: []gax.CallOption{},
 Unassign: []gax.CallOption{},
 EnumerateLicensedUsers: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
 }
}

// internalLicenseManagementClient is an interface that defines the methods available from Cloud Commerce
// Consumer Procurement API.
type internalLicenseManagementClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 GetLicensePool(context.Context, *procurementpb.GetLicensePoolRequest, ...gax.CallOption)
 (*procurementpb.LicensePool, error)
 UpdateLicensePool(context.Context, *procurementpb.UpdateLicensePoolRequest, ...gax.CallOption)
 (*procurementpb.LicensePool, error)
 Assign(context.Context, *procurementpb.AssignRequest, ...gax.CallOption) (*procurementpb.AssignResponse,
 error)
 Unassign(context.Context, *procurementpb.UnassignRequest,
 ...gax.CallOption) (*procurementpb.UnassignResponse, error)
 EnumerateLicensedUsers(context.Context, *procurementpb.EnumerateLicensedUsersRequest, ...gax.CallOption)
 *LicensedUserIterator
 GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
 (*longrunningpb.Operation, error)
}

// LicenseManagementClient is a client for interacting with Cloud Commerce Consumer Procurement API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// Service for managing licenses.
type LicenseManagementClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseManagementClient

 // The call options for this service.

```

```

CallOptions *LicenseManagementCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseManagementClient)
Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseManagementClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseManagementClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// GetLicensePool gets the license pool.
func (c *LicenseManagementClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 return c.internalClient.GetLicensePool(ctx, req, opts...)
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *LicenseManagementClient)
UpdateLicensePool(ctx context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 return c.internalClient.UpdateLicensePool(ctx, req, opts...)
}

// Assign assigns a license to a user.
func (c *LicenseManagementClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 return c.internalClient.Assign(ctx, req, opts...)
}

// Unassign unassigns a license from a user.
func (c *LicenseManagementClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest, opts

```

```

...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 return c.internalClient.Unassign(ctx, req, opts...)
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *LicenseManagementClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 return c.internalClient.EnumerateLicensedUsers(ctx,
 req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *LicenseManagementClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
 opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// licenseManagementGRPCClient is a client for interacting with Cloud Commerce Consumer Procurement API
// over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 // The gRPC API client.
 licenseManagementClient procurementpb.LicenseManagementServiceClient

 operationsClient longrunningpb.OperationsClient

 //
 // The x-goog-* metadata to be sent with each request.
 xGoogHeaders []string

 logger *slog.Logger
}

// NewLicenseManagementClient creates a new license management service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing licenses.
func NewLicenseManagementClient(ctx context.Context, opts ...option.ClientOption) (*LicenseManagementClient,
 error) {
 clientOpts := defaultLicenseManagementGRPCClientOptions()

```

```

if newLicenseManagementClientHook != nil {
 hookOpts, err := newLicenseManagementClientHook(ctx, clientHookParams{})
 if err != nil {
 return nil, err
 }
 clientOpts = append(clientOpts, hookOpts...)
}

connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
if err != nil {
 return nil, err
}
client := LicenseManagementClient{CallOptions: defaultLicenseManagementCallOptions()}

c := &licenseManagementGRPCClient{
 connPool:
 connPool,
 licenseManagementClient: procurementpb.NewLicenseManagementServiceClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
}
c.setGoogleClientInfo()

client.internalClient = c

return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *licenseManagementGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc",
 grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

```

```

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type licenseManagementRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseManagementClient
 CallOptions **LicenseManagementCallOptions

 logger *slog.Logger
}

// NewLicenseManagementRESTClient creates a new license management service rest client.
//
// Service for managing licenses.
func NewLicenseManagementRESTClient(ctx context.Context,
 opts ...option.ClientOption) (*LicenseManagementClient, error) {
 clientOpts := append(defaultLicenseManagementRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseManagementRESTCallOptions()
 c := &licenseManagementRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseManagementClient{internalClient: c, CallOptions: callOpts}, nil
}

```

```

func defaultLicenseManagementRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://cloudcommerceconsumerprocurement.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://cloudcommerceconsumerprocurement.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://cloudcommerceconsumerprocurement.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://cloudcommerceconsumerprocurement.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseManagementRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseManagementRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient
 = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseManagementRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *licenseManagementGRPCClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP

```

```

ool)], opts...)
var resp *procurementpb.LicensePool
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.GetLicensePool, req, settings.GRPC, c.logger,
"GetLicensePool")
 return err
},
opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

func (c *licenseManagementGRPCClient) UpdateLicensePool(ctx context.Context, req
*procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).Upda
teLicensePool)], opts...)
 var resp *procurementpb.LicensePool
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.UpdateLicensePool, req, settings.GRPC, c.logger,
"UpdateLicensePool")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *licenseManagementGRPCClient)
Assign(ctx context.Context, req *procurementpb.AssignRequest, opts ...gax.CallOption)
(*procurementpb.AssignResponse, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
 var resp *procurementpb.AssignResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

var err error
resp, err = executeRPC(ctx, c.licenseManagementClient.Assign, req, settings.GRPC, c.logger, "Assign")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

func (c *licenseManagementGRPCClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
hds := []string{"x-goog-request-params",
fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
var resp *procurementpb.UnassignResponse
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.licenseManagementClient.Unassign, req, settings.GRPC, c.logger, "Unassign")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

func (c *licenseManagementGRPCClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts
=
append((*c.CallOptions).EnumerateLicensedUsers[0:len((*c.CallOptions).EnumerateLicensedUsers):len((*c.CallOptions).EnumerateLicensedUsers)], opts...)
it := &LicensedUserIterator{}
req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
resp := &procurementpb.EnumerateLicensedUsersResponse{}
if pageToken != "" {
req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {

```



```

 req.PageSize = math.MaxInt32
} else if pageSize != 0 {
 req.PageSize = int32(pageSize)
}
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.licenseManagementClient.EnumerateLicensedUsers, req, settings.GRPC, c.logger,
"EnumerateLicensedUsers")
 return err
}, opts...)
if err != nil {
 return nil, "", err
}

it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(),
nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *licenseManagementGRPCClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
 var resp *longrunningpb.Operation
 err := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")

```

```

 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

// GetLicensePool gets the license pool.
func (c *licenseManagementRESTClient) GetLicensePool(ctx context.Context, req
*procurementpb.GetLicensePoolRequest, opts ...gax.CallOption) (*procurementpb.LicensePool, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx,
 hds...)
 opts =
 append((*c.CallOptions).GetLicensePool[0:len((*c.CallOptions).GetLicensePool):len((*c.CallOptions).GetLicenseP
ool)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &procurementpb.LicensePool{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLicensePool")
 if err != nil {
 return err
 }
 })
 if e != nil {
 return nil, e
 }
 return resp, nil
}

```

```

}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// UpdateLicensePool updates the license pool if one exists for this Order.
func (c *licenseManagementRESTClient) UpdateLicensePool(ctx
context.Context, req *procurementpb.UpdateLicensePoolRequest, opts ...gax.CallOption)
(*procurementpb.LicensePool, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetLicensePool()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetLicensePool().GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 params.Add("updateMask", string(field[1:len(field)-1]))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "license_pool.name",
url.QueryEscape(req.GetLicensePool().GetName()))}

 hds = append(c.xGoogHeaders, hds...)

```

```

hds = append(hds,
"Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).UpdateLicensePool[0:len((*c.CallOptions).UpdateLicensePool):len((*c.CallOptions).UpdateLicensePool)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &procurementpb.LicensePool{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateLicensePool")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// Assign assigns a license to a user.

```

func
(c *licenseManagementRESTClient) Assign(ctx context.Context, req *procurementpb.AssignRequest, opts
...gax.CallOption) (*procurementpb.AssignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

```

```

baseUrl, err := url.Parse(c.endpoint)

```

```

if err != nil {

```

```

return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v:assign", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Assign[0:len((*c.CallOptions).Assign):len((*c.CallOptions).Assign)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial:
true, DiscardUnknown: true}
resp := &procurementpb.AssignResponse{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Assign")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// Unassign unassigns a license from a user.

```

```

func (c *licenseManagementRESTClient) Unassign(ctx context.Context, req *procurementpb.UnassignRequest,
opts ...gax.CallOption) (*procurementpb.UnassignResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err :=
 m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:unassign", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Unassign[0:len((*c.CallOptions).Unassign):len((*c.CallOptions).Unassign)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &procurementpb.UnassignResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
 err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Unassign")
 if err != nil {
 return err
 }

 if err := unmarshalOptions.Unmarshal(buf, resp); err != nil {
 return err
 }
 })
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// EnumerateLicensedUsers enumerates all users assigned a license.
func (c *licenseManagementRESTClient) EnumerateLicensedUsers(ctx context.Context, req
*procurementpb.EnumerateLicensedUsersRequest, opts ...gax.CallOption) *LicensedUserIterator {
 it := &LicensedUserIterator{}
 req = proto.Clone(req).(*procurementpb.EnumerateLicensedUsersRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*procurementpb.LicensedUser, string, error) {
 resp := &procurementpb.EnumerateLicensedUsersResponse{}
 if
 pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:enumerateLicensedUsers", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {

```

```

 if settings.Path != "" {
 baseUrl.Path
 }
 = settings.Path
}
httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
if err != nil {
 return err
}
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "EnumerateLicensedUsers")
if err != nil {
 return err
}
if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLicensedUsers(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetOperation is a utility method from
google.longrunning.Operations.
func (c *licenseManagementRESTClient) GetOperation(ctx context.Context, req
*longrunningpb.GetOperationRequest, opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)

```



```

if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil

```

```

}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var
 newLicensesClientHook clientHook

// LicensesCallOptions contains the retry settings for each method of LicensesClient.
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption

```

```

Insert []gax.CallOption
List []gax.CallOption
SetIamPolicy []gax.CallOption
TestIamPermissions []gax.CallOption
Update []gax.CallOption
}

func defaultLicensesRESTCallOptions() *LicensesCallOptions {
return &LicensesCallOptions{
Delete: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
Get: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier:
1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
GetIamPolicy: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
Insert: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
},
List: []gax.CallOption{
gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 100 * time.Millisecond,
Max: 60000 * time.Millisecond,
Multiplier: 1.30,
},

```

```

 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 TestIamPermissions(context.Context,
 *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
 Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The Licenses API.
type LicensesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicensesClient

 // The call options for this service.
 CallOptions *LicensesCallOptions
}

```

```

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed
// on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context, req
 *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts
 ...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)

```

```

(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available in the
// specified project. This method does not get any licenses that belong to other projects, including licenses attached to
// publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to
// make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
(*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
// resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
// party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type
licensesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // operationClient is used to call the operation-specific management service.

```

```

operationClient *GlobalOperationsClient

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing LicensesClient
CallOptions **LicensesCallOptions

logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()
 c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
 }
 opC, err := NewGlobalOperationsRESTClient(ctx, o...)
 if err != nil {
 return nil, err
 }
 c.operationClient = opC

 return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 }
}

```

```

internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
internaloption.WithDefaultUniverseDomain("googleapis.com"),
internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
internaloption.WithDefaultScopes(DefaultAuthScopes()...),
internaloption.EnableNewAuthLibrary(),
}
}

// setGoogleClientInfo sets the name and version of the application in
//
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
"x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
if err := c.operationClient.Close(); err != nil {
return err
}
return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
return nil
}

// Delete deletes the specified license. Caution This resource is intended for use
// only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

```



```

params := url.Values{}
if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 }
}

```

```

 },
 }
 return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error
 {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshalOptions.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 })
}

```

```

 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
 opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/getIamPolicy",
 req.GetProject(), req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if
 settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

```

```

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
 err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")

```

```

headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &compute.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err
:= http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil
}

```

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) List(ctx context.Context, req *compute.ListLicensesRequest, opts
...gax.CallOption) *LicenseIterator {
 it := &LicenseIterator{}

```

```

req = proto.Clone(req).(*computepb.ListLicensesRequest)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
 resp := &computepb.LicensesListResponse{}
 if pageToken != "" {
 req.PageToken = proto.String(pageToken)
 }
 if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
 } else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.Filter != nil {
 params.Add("filter",
 fmt.Sprintf("%v", req.GetFilter()))
 }
 if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
 }
 if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
 }
 if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }
 if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err

```

```

 }
 httpReq.Header
= headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace
images.
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Policy{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```



```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) TestIamPermissions(ctx
context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
tIamPermissions)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }
 })
}

```

```

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Update(ctx context.Context,
 req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
 if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds,
 "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)

```

```

unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return
op, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

```

// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go-grpc. DO NOT EDIT.
// versions:
// - protoc-gen-go-grpc v1.3.0
// - protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license_service.proto

package discoveryenginepb

import (
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 context "context"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
)

//
// This is a compile-time assertion to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
// Requires gRPC-Go v1.32.0 or later.
const _ = grpc.SupportPackageIsVersion7

const (
 UserLicenseService_ListUserLicenses_FullMethodName =
 "/google.cloud.discoveryengine.v1.UserLicenseService/ListUserLicenses"
 UserLicenseService_BatchUpdateUserLicenses_FullMethodName =
 "/google.cloud.discoveryengine.v1.UserLicenseService/BatchUpdateUserLicenses"
)

// UserLicenseServiceClient is the client API for UserLicenseService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
// https://pkg.go.dev/google.golang.org/grpc/?tab=doc#ClientConn.NewStream.
type UserLicenseServiceClient interface {
 // Lists the User Licenses.
 ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest, opts ...grpc.CallOption)
 (*ListUserLicensesResponse, error)
 // Updates the User License.
 // This
 // method is used for batch assign/unassign licenses to users.
 BatchUpdateUserLicenses(ctx context.Context, in *BatchUpdateUserLicensesRequest, opts ...grpc.CallOption)
 (*longrunningpb.Operation, error)
}

type userLicenseServiceClient struct {

```

```

cc grpc.ClientConnInterface
}

func NewUserLicenseServiceClient(cc grpc.ClientConnInterface) UserLicenseServiceClient {
 return &userLicenseServiceClient{cc}
}

func (c *userLicenseServiceClient) ListUserLicenses(ctx context.Context, in *ListUserLicensesRequest, opts
...grpc.CallOption) (*ListUserLicensesResponse, error) {
 out := new(ListUserLicensesResponse)
 err := c.cc.Invoke(ctx, UserLicenseService_ListUserLicenses_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *userLicenseServiceClient) BatchUpdateUserLicenses(ctx context.Context, in
*BatchUpdateUserLicensesRequest, opts ...grpc.CallOption) (*longrunningpb.Operation, error) {
 out := new(longrunningpb.Operation)
 err := c.cc.Invoke(ctx,
UserLicenseService_BatchUpdateUserLicenses_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

// UserLicenseServiceServer is the server API for UserLicenseService service.
// All implementations should embed UnimplementedUserLicenseServiceServer
// for forward compatibility
type UserLicenseServiceServer interface {
 // Lists the User Licenses.
 ListUserLicenses(context.Context, *ListUserLicensesRequest) (*ListUserLicensesResponse, error)
 // Updates the User License.
 // This method is used for batch assign/unassign licenses to users.
 BatchUpdateUserLicenses(context.Context, *BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error)
}

// UnimplementedUserLicenseServiceServer should be embedded to have forward compatible implementations.
type UnimplementedUserLicenseServiceServer struct {
}

func (UnimplementedUserLicenseServiceServer) ListUserLicenses(context.Context, *ListUserLicensesRequest)
(*ListUserLicensesResponse, error) {
 return
 nil, status.Errorf(codes.Unimplemented, "method ListUserLicenses not implemented")
}

```

```

func (UnimplementedUserLicenseServiceServer) BatchUpdateUserLicenses(context.Context,
*BatchUpdateUserLicensesRequest) (*longrunningpb.Operation, error) {
 return nil, status.Errorf(codes.Unimplemented, "method BatchUpdateUserLicenses not implemented")
}

// UnsafeUserLicenseServiceServer may be embedded to opt out of forward compatibility for this service.
// Use of this interface is not recommended, as added methods to UserLicenseServiceServer will
// result in compilation errors.
type UnsafeUserLicenseServiceServer interface {
 mustEmbedUnimplementedUserLicenseServiceServer()
}

func RegisterUserLicenseServiceServer(s grpc.ServiceRegistrar, srv UserLicenseServiceServer) {
 s.RegisterService(&UserLicenseService_ServiceDesc, srv)
}

func _UserLicenseService_ListUserLicenses_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{},
error) {
 in := new(ListUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: UserLicenseService_ListUserLicenses_FullMethodName,
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).ListUserLicenses(ctx, req.(*ListUserLicensesRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _UserLicenseService_BatchUpdateUserLicenses_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(BatchUpdateUserLicensesRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: UserLicenseService_BatchUpdateUserLicenses_FullMethodName,

```

```

}
handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(UserLicenseServiceServer).BatchUpdateUserLicenses(ctx, req.(*BatchUpdateUserLicensesRequest))
}
return interceptor(ctx, in, info, handler)
}

// UserLicenseService_ServiceDesc is the grpc.ServiceDesc for UserLicenseService service.
// It's only intended for direct use with grpc.RegisterService,
// and not to be introspected or modified (even as a copy)
var UserLicenseService_ServiceDesc = grpc.ServiceDesc{
 ServiceName: "google.cloud.discoveryengine.v1.UserLicenseService",
 HandlerType: (*UserLicenseServiceServer)(nil),
 Methods: []grpc.MethodDesc{
 {
 MethodName: "ListUserLicenses",
 Handler: _UserLicenseService_ListUserLicenses_Handler,
 },
 {
 MethodName: "BatchUpdateUserLicenses",
 Handler: _UserLicenseService_BatchUpdateUserLicenses_Handler,
 },
 },
 Streams: []grpc.StreamDesc{},
 Metadata: "google/cloud/discoveryengine/v1/user_license_service.proto",
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,



worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

```

// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
)

var
 newLicenseCodesClientHook clientHook

// LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
}

func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{

```

```

 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
},
http.StatusGatewayTimeout,
http.StatusServiceUnavailable)
}),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface {
 Close()
 error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return c.internalClient.Close()
}

//

```

```

 setGoogleClientInfo sets the name and version of the application in
 // the `x-goog-api-client` header passed on each request. Intended for
 // use by Google-written clients.
 func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
 }

 // Connection returns a connection to the API service.
 //
 // Deprecated: Connections are now pooled so this method does not always
 // return the same resource.
 func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
 }

 // Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
 // License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
 // Marketplace images.
 func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
 ...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
 }

 // TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
 // intended for use only by third-party partners who are creating Cloud Marketplace images.
 func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
 *computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
 }

 // Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
 // method calls.
 type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions

 logger *slog.Logger
 }

```

```

}

// NewLicenseCodesRESTClient
creates a new license codes rest client.
//
// The LicenseCodes API.
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicenseCodesRESTCallOptions()
 c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

//
setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

```

```

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return
 nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {
 return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.LicenseCode{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }

```



```

httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a
// caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/v1/projects/%v/global/licenseCodes/%v/testIamPermissions",
 req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds =
 append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)

```

```

 opts =
append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).TestIamPermissions)], opts...)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
 if e != nil {
 return nil, e
 }
 return resp, nil
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

package compute

```
import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1beta/computepb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
)

var
 newLicenseCodesClientHook clientHook

 // LicenseCodesCallOptions contains the retry settings for each method of LicenseCodesClient.
 type LicenseCodesCallOptions struct {
 Get []gax.CallOption
 TestIamPermissions []gax.CallOption
 }

 func defaultLicenseCodesRESTCallOptions() *LicenseCodesCallOptions {
 return &LicenseCodesCallOptions{
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 }
 }
```

```

// internalLicenseCodesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicenseCodesClient interface
{
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Get(context.Context, *computepb.GetLicenseCodeRequest, ...gax.CallOption) (*computepb.LicenseCode, error)
 TestIamPermissions(context.Context, *computepb.TestIamPermissionsLicenseCodeRequest, ...gax.CallOption)
 (*computepb.TestPermissionsResponse, error)
}

// LicenseCodesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The LicenseCodes API.
type LicenseCodesClient struct {
 // The internal transport-dependent client.
 internalClient internalLicenseCodesClient

 // The call options for this service.
 CallOptions *LicenseCodesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicenseCodesClient) Close() error {
 return
 c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *LicenseCodesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicenseCodesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

```

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicenseCodesClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 return c.internalClient.Get(ctx,
 req, opts...)
}
```

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```
func (c *LicenseCodesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}
```

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with method calls.

```
type licenseCodesRESTClient struct {
 // The http endpoint to connect to.
 endpoint string
```

```
 // The http client.
 httpClient *http.Client
```

```
 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string
```

```
 // Points back to the CallOptions field of the containing LicenseCodesClient
 CallOptions **LicenseCodesCallOptions
```

```
 logger *slog.Logger
}
```

```
// NewLicenseCodesRESTClient
creates a new license codes rest client.
//
```

```
// The LicenseCodes API.
```

```
func NewLicenseCodesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicenseCodesClient, error) {
 clientOpts := append(defaultLicenseCodesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }
}
```

```
callOpts := defaultLicenseCodesRESTCallOptions()
```

```

c := &licenseCodesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

return &LicenseCodesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicenseCodesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
}
}

//
// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licenseCodesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licenseCodesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
return
nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licenseCodesRESTClient) Connection() *grpc.ClientConn {

```

```

return nil
}

// Get return a specified license code. License codes are mirrored across all projects that have permissions to read the
// License Code. Caution This resource is intended for use only by third-party partners who are creating Cloud
// Marketplace images.
func (c *licenseCodesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseCodeRequest, opts
...gax.CallOption) (*computepb.LicenseCode, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v", req.GetProject(),
req.GetLicenseCode())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license_code", url.QueryEscape(req.GetLicenseCode()))}

 hds
 = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.LicenseCode{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 }, opts...)
}

```

```

if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that
// a caller has on the specified resource. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *licenseCodesRESTClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseCodeRequest, opts ...gax.CallOption)
(*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenseCodes/%v/testIamPermissions",
 req.GetProject(), req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds
 = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
 tIamPermissions)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)

```



```

httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1alpha/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

```

```

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.
 Permission_GROUP Permission_GranteeType = 2
 // Represents access to everyone. No extra information is required.
 Permission_EVERYONE Permission_GranteeType = 3
)

// Enum value maps for Permission_GranteeType.
var (
 Permission_GranteeType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2: "GROUP",
 3: "EVERYONE",
 }
 Permission_GranteeType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GranteeType) Enum() *Permission_GranteeType {
 p := new(Permission_GranteeType)
 *p = x
 return p
}

func (x Permission_GranteeType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

```

```

}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 0}
}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

```

```

)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service
// accounts, you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache

```

unknownFields protoimpl.UnknownFields

```
// Output only. Identifier. The permission name. A unique name will be
// generated on create. Examples:
//
// tunedModels/{tuned_model}/permissions/{permission}
// corpora/{corpus}/permissions/{permission}
//
// Output only.
Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
// Optional. Immutable. The type of the grantee.
GranteeType *Permission_GranteeType
`protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_GranteeType,oneof"
json:"grantee_type,omitempty"`
// Optional. Immutable. The email address of the user of group which this
// permission refers. Field is not set when permission's grantee type is
// EVERYONE.
EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1alpha.Permission_Role,oneof"
json:"role,omitempty"`
}
```

```
func (x *Permission) Reset() {
 *x = Permission{}
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}
```

```
func (x *Permission) String() string {
 return protoimpl.X.MessageStringOf(x)
}
```

```
func (*Permission) ProtoMessage() {}
```

```
func (x *Permission) ProtoReflect() protoreflect.Message
{
 mi := &file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 }
 return ms
}
```

```

}
return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
 if x != nil && x.GranteeType != nil {
 return *x.GranteeType
 }
 return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1alpha_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc = []byte{
 0x0a, 0x35, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
 0x6e, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x24, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,
 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x1a, 0x1f, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f,

```

0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19,  
0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75,  
0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x87, 0x05, 0x0a, 0x0a, 0x50, 0x65,  
0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65,  
0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x6c, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,  
0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3c, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,  
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68,  
0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61,  
0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05,  
0x48, 0x00, 0x52, 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65,  
0x65, 0x54, 0x79, 0x70, 0x65, 0x88,  
0x01, 0x01, 0x12, 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72,  
0x65, 0x73, 0x73, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41,  
0x05, 0x48, 0x01, 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73,  
0x73, 0x88, 0x01, 0x01, 0x12, 0x53, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01,  
0x28, 0x0e, 0x32, 0x35, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67,  
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,  
0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,  
0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02,  
0x52, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61,  
0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41,  
0x4e,  
0x54, 0x45, 0x45, 0x5f, 0x54, 0x59, 0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49,  
0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01,  
0x12, 0x09, 0x0a, 0x05, 0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45,  
0x56, 0x45, 0x52, 0x59, 0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c,  
0x65, 0x12, 0x14, 0x0a, 0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43,  
0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52,  
0x10, 0x01, 0x12, 0x0a, 0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a,  
0x0a, 0x06, 0x52, 0x45, 0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6,  
0x01, 0x0a, 0x2c, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e,  
0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73,  
0x2e, 0x63, 0x6f, 0x6d,  
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12,  
0x32, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75,  
0x6e, 0x65, 0x64, 0x5f, 0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69,  
0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69,  
0x6f, 0x6e, 0x7d, 0x12, 0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f,  
0x72, 0x70, 0x75, 0x73, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,  
0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b,  
0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72,  
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e,  
0x74, 0x65, 0x65, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61,  
0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73,

```

0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42, 0x9d, 0x01, 0x0a, 0x28, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65,
0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61,
0x42, 0x0f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74,
0x6f, 0x50, 0x01, 0x5a, 0x5e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61,
0x70, 0x69, 0x76, 0x31, 0x61, 0x6c, 0x70, 0x68, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61,
0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75,
0x61, 0x67,
0x65, 0x70, 0x62, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
)

```

```

func file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1alpha_permission_proto_rawDescData
}

```

```

var file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes
= []any{
 (Permission_GrantType)(0), // 0: google.ai.generativelanguage.v1alpha.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1alpha.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1alpha.Permission
}
var file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1alpha.Permission.grantee_type:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1alpha.Permission.role:type_name ->
 google.ai.generativelanguage.v1alpha.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

```



```

func init() { file_google_ai_generativelanguage_v1alpha_permission_proto_init()
}
func file_google_ai_generativelanguage_v1alpha_permission_proto_init() {
if File_google_ai_generativelanguage_v1alpha_permission_proto != nil {
return
}
file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes[0].OneofWrappers = []any{}
type x struct{}
out := protoimpl.TypeBuilder{
File: protoimpl.DescBuilder{
GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
RawDescriptor: file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc,
NumEnums: 2,
NumMessages: 1,
NumExtensions: 0,
NumServices: 0,
},
GoTypes: file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes,
DependencyIndexes: file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs,
EnumInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_enumTypes,
MessageInfos: file_google_ai_generativelanguage_v1alpha_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1alpha_permission_proto = out.File
file_google_ai_generativelanguage_v1alpha_permission_proto_rawDesc
= nil
file_google_ai_generativelanguage_v1alpha_permission_proto_goTypes = nil
file_google_ai_generativelanguage_v1alpha_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
//
// Code generated by protoc-gen-go-grpc. DO NOT EDIT.
// versions:
// - protoc-gen-go-grpc v1.3.0
// - protoc v4.25.7
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

```

package procurementpb

```
import (
 context "context"
 grpc "google.golang.org/grpc"
 codes "google.golang.org/grpc/codes"
 status "google.golang.org/grpc/status"
)
```

```
// This is a compile-time assertion
// to ensure that this generated file
// is compatible with the grpc package it is being compiled against.
// Requires gRPC-Go v1.32.0 or later.
const _ = grpc.SupportPackageIsVersion7
```

```
const (
 LicenseManagementService_GetLicensePool_FullMethodName =
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/GetLicensePool"
 LicenseManagementService_UpdateLicensePool_FullMethodName =
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/UpdateLicensePool"
 LicenseManagementService_Assign_FullMethodName =
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Assign"
 LicenseManagementService_Unassign_FullMethodName =
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/Unassign"
 LicenseManagementService_EnumerateLicensedUsers_FullMethodName =
 "/google.cloud.commerce.consumer.procurement.v1.LicenseManagementService/EnumerateLicensedUsers"
)
```

```
// LicenseManagementServiceClient
// is the client API for LicenseManagementService service.
//
// For semantics around ctx use and closing/ending streaming RPCs, please refer to
// https://pkg.go.dev/google.golang.org/grpc/?tab=doc#ClientConn.NewStream.
type LicenseManagementServiceClient interface {
 // Gets the license pool.
 GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(ctx context.Context, in *UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(ctx context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(ctx context.Context, in *UnassignRequest, opts ...grpc.CallOption) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(ctx context.Context, in *EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error)
}
```

```

type licenseManagementServiceClient struct {
 cc grpc.ClientConnInterface
}

func NewLicenseManagementServiceClient(cc grpc.ClientConnInterface) LicenseManagementServiceClient {
 return &licenseManagementServiceClient{cc}
}

func (c *licenseManagementServiceClient) GetLicensePool(ctx context.Context, in *GetLicensePoolRequest, opts
...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx, LicenseManagementService_GetLicensePool_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) UpdateLicensePool(ctx context.Context, in
*UpdateLicensePoolRequest, opts ...grpc.CallOption) (*LicensePool, error) {
 out := new(LicensePool)
 err := c.cc.Invoke(ctx, LicenseManagementService_UpdateLicensePool_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Assign(ctx
context.Context, in *AssignRequest, opts ...grpc.CallOption) (*AssignResponse, error) {
 out := new(AssignResponse)
 err := c.cc.Invoke(ctx, LicenseManagementService_Assign_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

func (c *licenseManagementServiceClient) Unassign(ctx context.Context, in *UnassignRequest, opts
...grpc.CallOption) (*UnassignResponse, error) {
 out := new(UnassignResponse)
 err := c.cc.Invoke(ctx, LicenseManagementService_Unassign_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

func (c *licenseManagementServiceClient) EnumerateLicensedUsers(ctx context.Context, in
*EnumerateLicensedUsersRequest, opts ...grpc.CallOption) (*EnumerateLicensedUsersResponse, error) {
 out := new(EnumerateLicensedUsersResponse)
 err := c.cc.Invoke(ctx, LicenseManagementService_EnumerateLicensedUsers_FullMethodName, in, out, opts...)
 if err != nil {
 return nil, err
 }
 return out, nil
}

```

```

// LicenseManagementServiceServer
// is the server API for LicenseManagementService service.
// All implementations should embed UnimplementedLicenseManagementServiceServer
// for forward compatibility
type LicenseManagementServiceServer interface {
 // Gets the license pool.
 GetLicensePool(context.Context, *GetLicensePoolRequest) (*LicensePool, error)
 // Updates the license pool if one exists for this Order.
 UpdateLicensePool(context.Context, *UpdateLicensePoolRequest) (*LicensePool, error)
 // Assigns a license to a user.
 Assign(context.Context, *AssignRequest) (*AssignResponse, error)
 // Unassigns a license from a user.
 Unassign(context.Context, *UnassignRequest) (*UnassignResponse, error)
 // Enumerates all users assigned a license.
 EnumerateLicensedUsers(context.Context, *EnumerateLicensedUsersRequest)
(*EnumerateLicensedUsersResponse, error)
}

```

```

// UnimplementedLicenseManagementServiceServer should be embedded to have forward compatible
implementations.
type UnimplementedLicenseManagementServiceServer
struct {
}

```

```

func (UnimplementedLicenseManagementServiceServer) GetLicensePool(context.Context,
*GetLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method GetLicensePool not implemented")
}
func (UnimplementedLicenseManagementServiceServer) UpdateLicensePool(context.Context,
*UpdateLicensePoolRequest) (*LicensePool, error) {
 return nil, status.Errorf(codes.Unimplemented, "method UpdateLicensePool not implemented")
}
func (UnimplementedLicenseManagementServiceServer) Assign(context.Context, *AssignRequest)
(*AssignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Assign not implemented")
}
func (UnimplementedLicenseManagementServiceServer) Unassign(context.Context, *UnassignRequest)

```

```

(*UnassignResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method Unassign not implemented")
}

func (UnimplementedLicenseManagementServiceServer) EnumerateLicensedUsers(context.Context,
*EnumerateLicensedUsersRequest)
(*EnumerateLicensedUsersResponse, error) {
 return nil, status.Errorf(codes.Unimplemented, "method EnumerateLicensedUsers not implemented")
}

// UnsafeLicenseManagementServiceServer may be embedded to opt out of forward compatibility for this service.
// Use of this interface is not recommended, as added methods to LicenseManagementServiceServer will
// result in compilation errors.
type UnsafeLicenseManagementServiceServer interface {
 mustEmbedUnimplementedLicenseManagementServiceServer()
}

func RegisterLicenseManagementServiceServer(s grpc.ServiceRegistrar, srv LicenseManagementServiceServer) {
 s.RegisterService(&LicenseManagementService_ServiceDesc, srv)
}

func _LicenseManagementService_GetLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(GetLicensePoolRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx,
in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: LicenseManagementService_GetLicensePool_FullMethodName,
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).GetLicensePool(ctx, req.(*GetLicensePoolRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_UpdateLicensePool_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(UpdateLicensePoolRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, in)
 }
}

```

```

}
info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: LicenseManagementService_UpdateLicensePool_FullMethodName,
}
handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return
 srv.(LicenseManagementServiceServer).UpdateLicensePool(ctx, req.(*UpdateLicensePoolRequest))
}
return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Assign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in := new(AssignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Assign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: LicenseManagementService_Assign_FullMethodName,
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Assign(ctx, req.(*AssignRequest))
 }
 return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_Unassign_Handler(srv interface{}, ctx context.Context, dec func(interface{})
error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
 in
 := new(UnassignRequest)
 if err := dec(in); err != nil {
 return nil, err
 }
 if interceptor == nil {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, in)
 }
 info := &grpc.UnaryServerInfo{
 Server: srv,
 FullMethod: LicenseManagementService_Unassign_FullMethodName,
 }
 handler := func(ctx context.Context, req interface{}) (interface{}, error) {
 return srv.(LicenseManagementServiceServer).Unassign(ctx, req.(*UnassignRequest))
 }
}

```

```

return interceptor(ctx, in, info, handler)
}

func _LicenseManagementService_EnumerateLicensedUsers_Handler(srv interface{}, ctx context.Context, dec
func(interface{}) error, interceptor grpc.UnaryServerInterceptor) (interface{}, error) {
in := new(EnumerateLicensedUsersRequest)
if err := dec(in); err != nil {
return nil, err
}
if interceptor == nil {
return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx, in)
}
info := &grpc.UnaryServerInfo{
Server: srv,
FullMethod: LicenseManagementService_EnumerateLicensedUsers_FullMethodName,
}
handler
:= func(ctx context.Context, req interface{}) (interface{}, error) {
return srv.(LicenseManagementServiceServer).EnumerateLicensedUsers(ctx,
req.(*EnumerateLicensedUsersRequest))
}
return interceptor(ctx, in, info, handler)
}

// LicenseManagementService_ServiceDesc is the grpc.ServiceDesc for LicenseManagementService service.
// It's only intended for direct use with grpc.RegisterService,
// and not to be introspected or modified (even as a copy)
var LicenseManagementService_ServiceDesc = grpc.ServiceDesc{
ServiceName: "google.cloud.commerce.consumer.prourement.v1.LicenseManagementService",
HandlerType: (*LicenseManagementServiceServer)(nil),
Methods: []grpc.MethodDesc{
{
MethodName: "GetLicensePool",
Handler: _LicenseManagementService_GetLicensePool_Handler,
},
{
MethodName: "UpdateLicensePool",
Handler: _LicenseManagementService_UpdateLicensePool_Handler,
},
{
MethodName: "Assign",
Handler: _LicenseManagementService_Assign_Handler,
},
{
MethodName:
"Unassign",
Handler: _LicenseManagementService_Unassign_Handler,
},
},
}

```

```

{
 MethodName: "EnumerateLicensedUsers",
 Handler: _LicenseManagementService_EnumerateLicensedUsers_Handler,
},
},
Streams: []grpc.StreamDesc{},
Metadata: "google/cloud/commerce/consumer/procurement/v1/license_management_service.proto",
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```



```

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/computepb#ListLicensesRequest.
}
for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// +build ignore

```

```

package ignore
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

```

```

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

```

package discoveryengine_test

```

```

import (
 "context"

```

```

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

```

```

func ExampleNewUserLicenseClient() {
 ctx := context.Background()
 //

```

```

 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:

```

```

// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleNewUserLicenseRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseRESTClient(ctx)
 if
 err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleUserLicenseClient_BatchUpdateUserLicenses() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.BatchUpdateUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesReque

```

```

st.
}
op, err := c.BatchUpdateUserLicenses(ctx, req)
if err != nil {
 //
 TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleUserLicenseClient_ListUserLicenses() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &discoveryenginepb.ListUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
 }
 it := c.ListUserLicenses(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done
 {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.

```

```

// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*discoveryenginepb.ListUserLicensesResponse)
}
}

func ExampleUserLicenseClient_CancelOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.CancelOperationRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#CancelOperationRequest.
 }
 err = c.CancelOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleUserLicenseClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }
 resp,

```

```

err := c.GetOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleUserLicenseClient_ListOperations() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
 }
 it := c.ListOperations(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 //
 TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*longrunningpb.ListOperationsResponse)
 }
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");

```

```

// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/commerce/consumer/procurement/v1/license_management_service.proto

package procurementpb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 durationpb
 "google.golang.org/protobuf/types/known/durationpb"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Assignment protocol for a license pool.
type AssignmentProtocol struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // The type of assignment protocol.
 //
 // Types that are assignable to AssignmentType:
 //

```

```

// *AssignmentProtocol_ManualAssignmentType_
// *AssignmentProtocol_AutoAssignmentType_
AssignmentType isAssignmentProtocol_AssignmentType `protobuf_oneof:"assignment_type"`
}

func (x *AssignmentProtocol) Reset() {
 *x = AssignmentProtocol{}
 mi
 := &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol) ProtoMessage() {}

func (x *AssignmentProtocol) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0}
}

func (m *AssignmentProtocol) GetAssignmentType() isAssignmentProtocol_AssignmentType
{
 if m != nil {
 return m.AssignmentType
 }
 return nil
}

func (x *AssignmentProtocol) GetManualAssignmentType() *AssignmentProtocol_ManualAssignmentType {

```

```

if x, ok := x.GetAssignmentType().(*AssignmentProtocol_ManualAssignmentType_); ok {
 return x.ManualAssignmentType
}
return nil
}

func (x *AssignmentProtocol) GetAutoAssignmentType() *AssignmentProtocol_AutoAssignmentType {
 if x, ok := x.GetAssignmentType().(*AssignmentProtocol_AutoAssignmentType_); ok {
 return x.AutoAssignmentType
 }
 return nil
}

type isAssignmentProtocol_AssignmentType interface {
 isAssignmentProtocol_AssignmentType()
}

type AssignmentProtocol_ManualAssignmentType_ struct {
 // Allow manual assignments triggered by administrative operations only.
 ManualAssignmentType *AssignmentProtocol_ManualAssignmentType
 `protobuf:"bytes,2,opt,name=manual_assignment_type,json=manualAssignmentType,proto3,oneof"`
}

type AssignmentProtocol_AutoAssignmentType_ struct {
 // Allow automatic assignments
 // triggered by data plane operations.
 AutoAssignmentType *AssignmentProtocol_AutoAssignmentType
 `protobuf:"bytes,3,opt,name=auto_assignment_type,json=autoAssignmentType,proto3,oneof"`
}

func (*AssignmentProtocol_ManualAssignmentType_) isAssignmentProtocol_AssignmentType() {}

func (*AssignmentProtocol_AutoAssignmentType_) isAssignmentProtocol_AssignmentType() {}

// A license pool represents a pool of licenses that can be assigned to users.
type LicensePool struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Identifier. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Required. Assignment protocol for the license pool.
 LicenseAssignmentProtocol *AssignmentProtocol
 `protobuf:"bytes,2,opt,name=license_assignment_protocol,json=licenseAssignmentProtocol,proto3"
 json:"license_assignment_protocol,omitempty"`
 //

```



Output only. Licenses count that are available to be assigned.

AvailableLicenseCount int32

`protobuf:"varint,3,opt,name=available\_license\_count,json=availableLicenseCount,proto3"

json:"available\_license\_count,omitempty"``

// Output only. Total number of licenses in the pool.

TotalLicenseCount int32 `protobuf:"varint,4,opt,name=total\_license\_count,json=totalLicenseCount,proto3"

json:"total\_license\_count,omitempty"``

}

func (x \*LicensePool) Reset() {

  \*x = LicensePool{}

  mi :=

  &file\_google\_cloud\_commerce\_consumer\_procurement\_v1\_license\_management\_service\_proto\_msgTypes[1]

  ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

  ms.StoreMessageInfo(mi)

}

func (x \*LicensePool) String() string {

  return protoimpl.X.MessageStringOf(x)

}

func (\*LicensePool) ProtoMessage() {}

func (x \*LicensePool) ProtoReflect() protoreflect.Message {

  mi :=

  &file\_google\_cloud\_commerce\_consumer\_procurement\_v1\_license\_management\_service\_proto\_msgTypes[1]

  if x != nil {

    ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

    if

    ms.LoadMessageInfo() == nil {

      ms.StoreMessageInfo(mi)

    }

    return ms

  }

  return mi.MessageOf(x)

}

// Deprecated: Use LicensePool.ProtoReflect.Descriptor instead.

func (\*LicensePool) Descriptor() ([]byte, []int) {

  return

  file\_google\_cloud\_commerce\_consumer\_procurement\_v1\_license\_management\_service\_proto\_rawDescGZIP(),

  []int{1}

}

func (x \*LicensePool) GetName() string {

  if x != nil {

    return x.Name

  }

```

return ""
}

func (x *LicensePool) GetLicenseAssignmentProtocol() *AssignmentProtocol {
 if x != nil {
 return x.LicenseAssignmentProtocol
 }
 return nil
}

func (x *LicensePool) GetAvailableLicenseCount() int32 {
 if x != nil {
 return x.AvailableLicenseCount
 }
 return 0
}

func (x *LicensePool) GetTotalLicenseCount() int32 {
 if x != nil {
 return x.TotalLicenseCount
 }
 return 0
}

// Request message for getting a license pool.
type GetLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields
 protoimpl.UnknownFields

 // Required. The name of the license pool to get.
 // Format: `billingAccounts/{billing_account}/orders/{order}/licensePool`
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
}

func (x *GetLicensePoolRequest) Reset() {
 *x = GetLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *GetLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

```

```

func (*GetLicensePoolRequest) ProtoMessage() {}

func (x *GetLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[2]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated:
// Use GetLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*GetLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{2}
}

func (x *GetLicensePoolRequest) GetName() string {
 if x != nil {
 return x.Name
 }
 return ""
}

// Request message for updating a license pool.
type UpdateLicensePoolRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The license pool to update.
 //
 // The license pool's name field is used to identify the license pool to
 // update. Format:
 // `billingAccounts/{billing_account}/orders/{order}/licensePool`.
 LicensePool *LicensePool `protobuf:"bytes,1,opt,name=license_pool,json=licensePool,proto3"
 json:"license_pool,omitempty"`
 // Required. The list of fields to update.
 UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
 json:"update_mask,omitempty"`
}

```

```

func (x *UpdateLicensePoolRequest) Reset() {
 *x = UpdateLicensePoolRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UpdateLicensePoolRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UpdateLicensePoolRequest) ProtoMessage() {}

func (x *UpdateLicensePoolRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use UpdateLicensePoolRequest.ProtoReflect.Descriptor instead.
func (*UpdateLicensePoolRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{3}
}

func (x *UpdateLicensePoolRequest) GetLicensePool() *LicensePool {
 if x != nil {
 return x.LicensePool
 }
 return nil
}

func (x *UpdateLicensePoolRequest) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

```

```

// Request message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *AssignRequest) Reset() {
 *x = AssignRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignRequest) ProtoMessage() {}

func (x *AssignRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[4]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignRequest.ProtoReflect.Descriptor instead.
func (*AssignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),

```

```

[]int{4}
}

func (x *AssignRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *AssignRequest) GetUsernames() []string {
 if x != nil {
 {
 return x.Usernames
 }
 }
 return nil
}

// Response message for
//
[LicenseManagementService.Assign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign].
type AssignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *AssignResponse) Reset() {
 *x = AssignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignResponse) ProtoMessage() {}

func (x *AssignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[5]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {

```

```

 ms.StoreMessageInfo(mi)
}
return
ms
}
return mi.MessageOf(x)
}

// Deprecated: Use AssignResponse.ProtoReflect.Descriptor instead.
func (*AssignResponse) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{5}
}

// Request message for
//
[LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementSer
vice.Unassign].
type UnassignRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Required. Username.
 // Format: `name@domain.com`.
 Usernames []string `protobuf:"bytes,2,rep,name=usernames,proto3" json:"usernames,omitempty"`
}

func (x *UnassignRequest) Reset() {
 *x = UnassignRequest{}
 mi :=
&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UnassignRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignRequest) ProtoMessage() {}

func (x *UnassignRequest) ProtoReflect() protoreflect.Message {
 mi :=

```

```

&file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[6]
if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use UnassignRequest.ProtoReflect.Descriptor instead.
func (*UnassignRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{6}
}

func (x *UnassignRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *UnassignRequest) GetUsernames() []string {
 if x != nil {
 return x.Usernames
 }
 return
 nil
}

// Response message for
//
// [LicenseManagementService.Unassign][google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign].
type UnassignResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func (x *UnassignResponse) Reset() {
 *x = UnassignResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))

```



```

ms.StoreMessageInfo(mi)
}

func (x *UnassignResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UnassignResponse) ProtoMessage() {}

func (x *UnassignResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[7]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return
 mi.MessageOf(x)
}

// Deprecated: Use UnassignResponse.ProtoReflect.Descriptor instead.
func (*UnassignResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{7}
}

// Request message for
//
// [LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
// ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. License pool name.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. The maximum number of users to return. The service may return
 // fewer than this value.
 PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3" json:"page_size,omitempty"`
 // Optional. A page token, received from a previous `EnumerateLicensedUsers`
 //
 // call. Provide this to retrieve the subsequent page.
 PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
}

```

```

}

func (x *EnumerateLicensedUsersRequest) Reset() {
 *x = EnumerateLicensedUsersRequest{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *EnumerateLicensedUsersRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*EnumerateLicensedUsersRequest) ProtoMessage() {}

func (x *EnumerateLicensedUsersRequest) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[8]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use EnumerateLicensedUsersRequest.ProtoReflect.Descriptor
// instead.
func (*EnumerateLicensedUsersRequest) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{8}
}

func (x *EnumerateLicensedUsersRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *EnumerateLicensedUsersRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
}

```

```

return 0
}

func (x *EnumerateLicensedUsersRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

// A licensed user.
type LicensedUser struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Username.
 // Format: `name@domain.com`.
 Username string `protobuf:"bytes,1,opt,name=username,proto3" json:"username,omitempty"`
 // Output only. Timestamp when the license was assigned.
 AssignTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=assign_time,json=assignTime,proto3"
 json:"assign_time,omitempty"`
 // Output only. Timestamp when the license was recently used. This may not be
 // the most recent usage time, and will be updated regularly (within 24
 // hours).
 RecentUsageTime *timestamppb.Timestamp
 `protobuf:"bytes,3,opt,name=recent_usage_time,json=recentUsageTime,proto3"
 json:"recent_usage_time,omitempty"`
}

func (x *LicensedUser) Reset() {
 *x = LicensedUser{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *LicensedUser) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*LicensedUser) ProtoMessage() {}

func (x *LicensedUser) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[9]
 if x != nil {

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo()
== nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use LicensedUser.ProtoReflect.Descriptor instead.
func (*LicensedUser) Descriptor() ([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{9}
}

func (x *LicensedUser) GetUsername() string {
 if x != nil {
 return x.Username
 }
 return ""
}

func (x *LicensedUser) GetAssignTime() *timestamppb.Timestamp {
 if x != nil {
 return x.AssignTime
 }
 return nil
}

func (x *LicensedUser) GetRecentUsageTime() *timestamppb.Timestamp {
 if x != nil {
 return x.RecentUsageTime
 }
 return nil
}

// Response message for
//
[LicenseManagementService.EnumerateLicensedUsers][google.cloud.commerce.consumer.procurement.v1.License
ManagementService.EnumerateLicensedUsers].
type EnumerateLicensedUsersResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 //

```

The list of licensed users.

```
LicensedUsers []*LicensedUser `protobuf:"bytes,1,rep,name=licensed_users,json=licensedUsers,proto3"
json:"licensed_users,omitempty"`
// A token that can be sent as `page_token` to retrieve the next page.
// If this field is omitted, there are no subsequent pages.
NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
json:"next_page_token,omitempty"`
}
```

```
func (x *EnumerateLicensedUsersResponse) Reset() {
 *x = EnumerateLicensedUsersResponse{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}
```

```
func (x *EnumerateLicensedUsersResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}
```

```
func (*EnumerateLicensedUsersResponse) ProtoMessage() {}
```

```
func (x *EnumerateLicensedUsersResponse) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[10]
 if
 x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}
```

// Deprecated: Use EnumerateLicensedUsersResponse.ProtoReflect.Descriptor instead.

```
func (*EnumerateLicensedUsersResponse) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{10}
}
```

```
func (x *EnumerateLicensedUsersResponse) GetLicensedUsers() []*LicensedUser {
 if x != nil {
 return x.LicensedUsers
 }
}
```

```

return nil
}

func (x *EnumerateLicensedUsersResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
 return ""
}

// Allow manual assignments triggered by administrative operations only.
type AssignmentProtocol_ManualAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields
}

func
(x *AssignmentProtocol_ManualAssignmentType) Reset() {
 *x = AssignmentProtocol_ManualAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *AssignmentProtocol_ManualAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_ManualAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_ManualAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[11]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use AssignmentProtocol_ManualAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_ManualAssignmentType) Descriptor()

```

```

([]byte, []int) {
 return
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
[]int{0, 0}
}

// Configuration for automatic assignments handled by data plane operations.
type AssignmentProtocol_AutoAssignmentType struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Optional. The time to live for an inactive license. After this time has
 // passed, the license will be automatically unassigned from the user. Must
 // be at least 7 days, if set. If unset, the license will never expire.
 InactiveLicenseTtl *durationpb.Duration
 `protobuf:"bytes,1,opt,name=inactive_license_ttl,json=inactiveLicenseTtl,proto3"
 json:"inactive_license_ttl,omitempty"`
}

func (x *AssignmentProtocol_AutoAssignmentType) Reset() {
 *x = AssignmentProtocol_AutoAssignmentType{}
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func
(x *AssignmentProtocol_AutoAssignmentType) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*AssignmentProtocol_AutoAssignmentType) ProtoMessage() {}

func (x *AssignmentProtocol_AutoAssignmentType) ProtoReflect() protoreflect.Message {
 mi :=
 &file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[12]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

// Deprecated: Use AssignmentProtocol_AutoAssignmentType.ProtoReflect.Descriptor instead.
func (*AssignmentProtocol_AutoAssignmentType) Descriptor() ([]byte, []int) {
 return
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP(),
 []int{0, 1}
}

func (x *AssignmentProtocol_AutoAssignmentType) GetInactiveLicenseTtl() *durationpb.Duration
{
 if x != nil {
 return x.InactiveLicenseTtl
 }
 return nil
}

var File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto
protorelect.FileDescriptor

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc =
[]byte{
 0x0a, 0x4e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x76, 0x31, 0x2f,
 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x6d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x73, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x12, 0x2d, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63,
 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x63,
 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x1a,
 0x1c, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f,
 0x74, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f,
 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f,
 0x61, 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x1e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x64, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,
 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e,
 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc3, 0x03, 0x0a, 0x12, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,
 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x8e, 0x01, 0x0a,
 0x16, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,
 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x56, 0x2e,
 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d,

```



0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70,  
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41,  
0x73,  
0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c,  
0x2e, 0x4d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x54, 0x79, 0x70, 0x65, 0x48, 0x00, 0x52, 0x14, 0x6d, 0x61, 0x6e, 0x75, 0x61, 0x6c, 0x41,  
0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x88, 0x01,  
0x0a, 0x14, 0x61, 0x75, 0x74, 0x6f, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e,  
0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x54, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d,  
0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72,  
0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x2e,  
0x41, 0x75, 0x74, 0x6f,  
0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79,  
0x70, 0x65, 0x48, 0x00, 0x52, 0x12, 0x61, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x1a, 0x16, 0x0a, 0x14, 0x4d, 0x61, 0x6e, 0x75,  
0x61, 0x6c, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x54, 0x79, 0x70, 0x65,  
0x1a, 0x66, 0x0a, 0x12, 0x41, 0x75, 0x74, 0x6f, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65,  
0x6e, 0x74, 0x54, 0x79, 0x70, 0x65, 0x12, 0x50, 0x0a, 0x14, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69,  
0x76, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x74, 0x74, 0x6c, 0x18, 0x01,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x19, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x44, 0x75, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x42,  
0x03, 0xe0, 0x41, 0x01, 0x52, 0x12, 0x69, 0x6e, 0x61, 0x63, 0x74, 0x69, 0x76, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x54, 0x74, 0x6c, 0x42,  
0x11, 0x0a, 0x0f, 0x61, 0x73, 0x73, 0x69,  
0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x74, 0x79, 0x70, 0x65, 0x22, 0xbe, 0x03, 0x0a, 0x0b,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x17, 0x0a, 0x04, 0x6e,  
0x61, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04,  
0x6e, 0x61, 0x6d, 0x65, 0x12, 0x86, 0x01, 0x0a, 0x1b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x5f, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x63, 0x6f, 0x6c, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x41, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x42,  
0x03, 0xe0,  
0x41, 0x02, 0x52, 0x19, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67,  
0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x63, 0x6f, 0x6c, 0x12, 0x3b, 0x0a,  
0x17, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x15, 0x61, 0x76, 0x61, 0x69, 0x6c, 0x61, 0x62, 0x6c, 0x65, 0x4c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x33, 0x0a, 0x13, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e,  
0x74, 0x18, 0x04, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x11, 0x74, 0x6f,  
0x74, 0x61, 0x6c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x3a,  
0x9a, 0x01, 0xea, 0x41, 0x96, 0x01, 0x0a, 0x3b, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d,  
0x6d, 0x65, 0x72,

0x63, 0x65, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61,  
0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x12, 0x3c, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f,  
0x75, 0x6e, 0x74, 0x73, 0x2f, 0x7b, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x5f, 0x61, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x7d, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x7b, 0x6f,  
0x72, 0x64, 0x65, 0x72, 0x7d, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f,  
0x6c, 0x2a, 0x0c, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x73, 0x32,  
0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x30, 0x0a, 0x15,  
0x47, 0x65, 0x74, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x17, 0x0a,  
0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x22, 0xc0,  
0x01, 0x0a, 0x18, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x62, 0x0a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x0b, 0x32, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x0b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12,  
0x40, 0x0a, 0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73,  
0x6b, 0x18, 0x02,  
0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b,  
0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73,  
0x6b, 0x22, 0x4f, 0x0a, 0x0d, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65,  
0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12,  
0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03,  
0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d,  
0x65, 0x73, 0x22, 0x10, 0x0a, 0x0e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70,  
0x6f, 0x6e, 0x73, 0x65, 0x22, 0x51, 0x0a, 0x0f, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e,  
0x52, 0x65,  
0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e,  
0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x06, 0x70, 0x61,  
0x72, 0x65, 0x6e, 0x74, 0x12, 0x21, 0x0a, 0x09, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52, 0x09, 0x75, 0x73,  
0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x22, 0x12, 0x0a, 0x10, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x82, 0x01, 0x0a, 0x1d,  
0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x1b, 0x0a,  
0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0,  
0x41, 0x02, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20, 0x0a, 0x09, 0x70, 0x61,  
0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65,  
0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03, 0xe0,  
0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a, 0x0a,  
0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09,

0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e,  
0x22, 0xb9, 0x01, 0x0a, 0x0c, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x12, 0x1a, 0x0a, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01, 0x20,  
0x01, 0x28, 0x09, 0x52, 0x08, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x12, 0x40, 0x0a,  
0x0b, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,  
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x54,  
0x69, 0x6d, 0x65, 0x12,  
0x4b, 0x0a, 0x11, 0x72, 0x65, 0x63, 0x65, 0x6e, 0x74, 0x5f, 0x75, 0x73, 0x61, 0x67, 0x65, 0x5f,  
0x74, 0x69, 0x6d, 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x0f, 0x72, 0x65, 0x63,  
0x65, 0x6e, 0x74, 0x55, 0x73, 0x61, 0x67, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x22, 0xac, 0x01, 0x0a,  
0x1e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12,  
0x62, 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x5f, 0x75, 0x73, 0x65, 0x72,  
0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,  
0x63,  
0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65,  
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64,  
0x55, 0x73, 0x65, 0x72, 0x52, 0x0d, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73,  
0x65, 0x72, 0x73, 0x12, 0x26, 0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65,  
0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65,  
0x78, 0x74, 0x50, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x32, 0xc3, 0x0a, 0x0a, 0x18,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd4, 0x01, 0x0a, 0x0e, 0x47, 0x65, 0x74,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x44, 0x2e, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65,  
0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f,  
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x47, 0x65, 0x74, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x1a, 0x3a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,  
0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d,  
0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x40, 0xda,  
0x41, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x33, 0x12, 0x31, 0x2f, 0x76,  
0x31, 0x2f, 0x7b, 0x6e, 0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41,  
0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73,  
0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50,  
0x6f, 0x6f, 0x6c, 0x7d, 0x12,  
0x89, 0x02, 0x0a, 0x11, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x12, 0x47, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f,  
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65,  
0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,

0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3a,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f,  
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e,  
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x4c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x22, 0x6f, 0xda, 0x41, 0x18, 0x6c,  
0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2c, 0x75, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4e, 0x3a, 0x0c, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x32, 0x3e, 0x2f, 0x76, 0x31,  
0x2f, 0x7b, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x70, 0x6f, 0x6f, 0x6c, 0x2e, 0x6e,  
0x61, 0x6d, 0x65, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75,  
0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c,  
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x12, 0xdf, 0x01, 0x0a, 0x06,  
0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63,  
0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d,  
0x65, 0x6e, 0x74, 0x2e, 0x76,  
0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x71,  
0x75, 0x65, 0x73, 0x74, 0x1a, 0x3d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e,  
0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,  
0x74, 0x2e, 0x76, 0x31, 0x2e, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f,  
0x6e, 0x73, 0x65, 0x22, 0x58, 0xda, 0x41, 0x10, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75,  
0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65, 0x73, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x3f, 0x3a, 0x01,  
0x2a, 0x22, 0x3a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62,  
0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a,  
0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x12, 0xe7, 0x01,  
0x0a, 0x08, 0x55, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x3e, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x3f, 0x2e, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72,  
0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,  
0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x6e, 0x61, 0x73, 0x73,  
0x69, 0x67, 0x6e, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22, 0x5a, 0xda, 0x41, 0x10,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x2c, 0x75, 0x73, 0x65, 0x72, 0x6e, 0x61, 0x6d, 0x65,  
0x73,  
0x82, 0xd3, 0xe4, 0x93, 0x02, 0x41, 0x3a, 0x01, 0x2a, 0x22, 0x3c, 0x2f, 0x76, 0x31, 0x2f, 0x7b,  
0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69, 0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63,  
0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f,  
0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x75,  
0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x12, 0x92, 0x02, 0x0a, 0x16, 0x45, 0x6e, 0x75, 0x6d,  
0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65,  
0x72, 0x73, 0x12, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75,

```

0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e,
0x76, 0x31, 0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65,
0x6e, 0x73, 0x65,
0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74,
0x1a, 0x4d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,
0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,
0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31,
0x2e, 0x45, 0x6e, 0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,
0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x22,
0x5b, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4c,
0x12, 0x4a, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x62, 0x69,
0x6c, 0x6c, 0x69, 0x6e, 0x67, 0x41, 0x63, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x73, 0x2f, 0x2a, 0x2f,
0x6f, 0x72, 0x64, 0x65, 0x72, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x50, 0x6f, 0x6f, 0x6c, 0x7d, 0x3a, 0x65, 0x6e,
0x75, 0x6d, 0x65, 0x72, 0x61, 0x74, 0x65, 0x4c,
0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x73, 0x1a, 0x63, 0xca, 0x41,
0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x63, 0x6f,
0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e,
0x74, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,
0xd2, 0x41, 0x2e, 0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75,
0x74, 0x68, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72,
0x6d, 0x42, 0xbe, 0x02, 0x0a, 0x31, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x63, 0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e,
0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x63,
0x75, 0x72, 0x65,
0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x76, 0x31, 0x42, 0x1d, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x4d, 0x61, 0x6e, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63,
0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x53, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x63, 0x6f,
0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x2f, 0x63, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2f,
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2f, 0x61, 0x70, 0x69, 0x76,
0x31, 0x2f, 0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0x3b,
0x70, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x70, 0x62, 0xaa, 0x02, 0x2d,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x43, 0x6f, 0x6d,
0x6d, 0x65, 0x72, 0x63, 0x65, 0x2e, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x2e, 0x50,
0x72, 0x6f,
0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x2d,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x43, 0x6f, 0x6d,
0x6d, 0x65, 0x72, 0x63, 0x65, 0x5c, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65, 0x72, 0x5c, 0x50,
0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x32,
0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x43,
0x6f, 0x6d, 0x6d, 0x65, 0x72, 0x63, 0x65, 0x3a, 0x3a, 0x43, 0x6f, 0x6e, 0x73, 0x75, 0x6d, 0x65,
0x72, 0x3a, 0x3a, 0x50, 0x72, 0x6f, 0x63, 0x75, 0x72, 0x65, 0x6d, 0x65, 0x6e, 0x74, 0x3a, 0x3a,
0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```
var (
```

```

file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce
sync.Once
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc
)

func
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescGZIP()
[]byte {
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescOnce.Do(f
unc() {
file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_commerce_consumer_procurement_v1_license_management_servi
ce_proto_rawDescData)
})
return file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDescData
}

var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 13)
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes =
[]any{
(*AssignmentProtocol)(nil), // 0:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
(*LicensePool)(nil),
// 1: google.cloud.commerce.consumer.procurement.v1.LicensePool
(*GetLicensePoolRequest)(nil), // 2:
google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
(*UpdateLicensePoolRequest)(nil), // 3:
google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
(*AssignRequest)(nil), // 4: google.cloud.commerce.consumer.procurement.v1.AssignRequest
(*AssignResponse)(nil), // 5: google.cloud.commerce.consumer.procurement.v1.AssignResponse
(*UnassignRequest)(nil), // 6: google.cloud.commerce.consumer.procurement.v1.UnassignRequest
(*UnassignResponse)(nil), // 7: google.cloud.commerce.consumer.procurement.v1.UnassignResponse
(*EnumerateLicensedUsersRequest)(nil), // 8:
google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
(*LicensedUser)(nil), // 9: google.cloud.commerce.consumer.procurement.v1.LicensedUser
(*EnumerateLicensedUsersResponse)(nil),
// 10: google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
(*AssignmentProtocol_ManualAssignmentType)(nil), // 11:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
(*AssignmentProtocol_AutoAssignmentType)(nil), // 12:
google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
(*fieldmaskpb.FieldMask)(nil), // 13: google.protobuf.FieldMask
(*timestamppb.Timestamp)(nil), // 14: google.protobuf.Timestamp
(*durationpb.Duration)(nil), // 15: google.protobuf.Duration
}
var file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs =

```

```

[]int32{
 11, // 0:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.manual_assignment_type:type_name ->
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.ManualAssignmentType
 12,
 // 1: google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.auto_assignment_type:type_name ->
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType
 0, // 2: google.cloud.commerce.consumer.procurement.v1.LicensePool.license_assignment_protocol:type_name ->
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol
 1, // 3: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.license_pool:type_name ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 13, // 4: google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest.update_mask:type_name ->
 google.protobuf.FieldMask
 14, // 5: google.cloud.commerce.consumer.procurement.v1.LicensedUser.assign_time:type_name ->
 google.protobuf.Timestamp
 14, // 6: google.cloud.commerce.consumer.procurement.v1.LicensedUser.recent_usage_time:type_name ->
 google.protobuf.Timestamp
 9, // 7:
 google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse.licensed_users:type_name
 -> google.cloud.commerce.consumer.procurement.v1.LicensedUser
 15, // 8:
 google.cloud.commerce.consumer.procurement.v1.AssignmentProtocol.AutoAssignmentType.inactive_license_ttl:t
 ype_name -> google.protobuf.Duration
 2, // 9: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:input_type -
 > google.cloud.commerce.consumer.procurement.v1.GetLicensePoolRequest
 3, // 10:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:input_type ->
 google.cloud.commerce.consumer.procurement.v1.UpdateLicensePoolRequest
 4, // 11: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:input_type ->
 google.cloud.commerce.consumer.procurement.v1.AssignRequest
 6, // 12: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:input_type ->
 google.cloud.commerce.consumer.procurement.v1.UnassignRequest
 8, // 13:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:input_ty
 p
 e
 -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersRequest
 1, // 14:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.GetLicensePool:output_type ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 1, // 15:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.UpdateLicensePool:output_type ->
 google.cloud.commerce.consumer.procurement.v1.LicensePool
 5, // 16: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Assign:output_type ->
 google.cloud.commerce.consumer.procurement.v1.AssignResponse
 7, // 17: google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.Unassign:output_type ->
 google.cloud.commerce.consumer.procurement.v1.UnassignResponse
 10, // 18:
 google.cloud.commerce.consumer.procurement.v1.LicenseManagementService.EnumerateLicensedUsers:output_ty

```

```

pe -> google.cloud.commerce.consumer.procurement.v1.EnumerateLicensedUsersResponse
14,
// [14:19] is the sub-list for method output_type
9, // [9:14] is the sub-list for method input_type
9, // [9:9] is the sub-list for extension type_name
9, // [9:9] is the sub-list for extension extendee
0, // [0:9] is the sub-list for field type_name
}

func init() {
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init()
}
func file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_init() {
 if File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto != nil {
 return
 }
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes[0].Oneof
 Wrappers = []any{
 (*AssignmentProtocol_ManualAssignmentType_)(nil),
 (*AssignmentProtocol_AutoAssignmentType_)(nil),
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc,
 NumEnums:
 0,
 NumMessages: 13,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes,
 DependencyIndexes:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs,
 MessageInfos:
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_msgTypes,
 }.Build()
 File_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto = out.File
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_rawDesc = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_goTypes = nil
 file_google_cloud_commerce_consumer_procurement_v1_license_management_service_proto_depIdxs = nil
}

// Copyright 2023 Google LLC
//

```



```

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package licensemanager_test

import (
 "context"

 licensemanager "cloud.google.com/go/licensemanager/apiv1"
 licensemanagerpb "cloud.google.com/go/licensemanager/apiv1/licensemanagerpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
 locationpb "google.golang.org/genproto/googleapis/cloud/location"
)

func
ExampleNewClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)

```

```

if err != nil {
 // TODO: Handle error.
}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleNewRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := licensemanager.NewRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleClient_AggregateUsage() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.AggregateUsageRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#AggregateUsageRequest.
 }
 it := c.AggregateUsage(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {

```

```

 break
}
if
err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp

// If you need to access the underlying RPC response,
// you can do so by casting the `Response` as below.
// Otherwise, remove this line. Only populated after
// first call to Next(). Not safe for concurrent access.
_ = it.Response.(*licensemanagerpb.AggregateUsageResponse)
}
}

func ExampleClient_CreateConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.CreateConfigurationRequest{
 // TODO: Fill request
 struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#CreateConfigurationRequest.
 }
 op, err := c.CreateConfiguration(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 resp, err := op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

```

```

func ExampleClient_DeactivateConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.DeactivateConfigurationRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#DeactivateConfigurationRequest.
 }
 op,
 err := c.DeactivateConfiguration(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 resp, err := op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

```

```

func ExampleClient_DeleteConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.DeleteConfigurationRequest{
 // TODO: Fill request struct fields.

```

```

// See
https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#DeleteConfigurationRequest.
}
op,
err := c.DeleteConfiguration(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleClient_GetConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.GetConfigurationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#GetConfigurationRequest.
 }
 resp, err := c.GetConfiguration(ctx,
 req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleClient_GetInstance() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options

```

```

c, err := licensemanager.NewClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &licensemanagerpb.GetInstanceRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#GetInstanceRequest.
}
resp, err := c.GetInstance(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

```

func ExampleClient_GetProduct() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &licensemanagerpb.GetProductRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#GetProductRequest.
}
resp, err := c.GetProduct(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

```

func ExampleClient_ListConfigurations() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to

```

```

work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := licensemanager.NewClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &licensemanagerpb.ListConfigurationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ListConfigurationsRequest.
}
it := c.ListConfigurations(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*licensemanagerpb.ListConfigurationsResponse)
}
}

func
ExampleClient_ListInstances() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &licensemanagerpb.ListInstancesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ListInstancesRequest.
}
it := c.ListInstances(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the
 // underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*licensemanagerpb.ListInstancesResponse)
}
}

```

```

func ExampleClient_ListProducts() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &licensemanagerpb.ListProductsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ListProductsRequest.
}
it
:= c.ListProducts(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
}

```



```

 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*licensemanagerpb.ListProductsResponse)
}
}

func ExampleClient_QueryConfigurationLicenseUsage() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO:
 Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.QueryConfigurationLicenseUsageRequest{
 // TODO: Fill request struct fields.
 // See
https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#QueryConfigurationLicenseUsageRequest.
 }
 resp, err := c.QueryConfigurationLicenseUsage(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleClient_ReactivateConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.

```

```

// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := licensemanager.NewClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &licensemanagerpb.ReactivateConfigurationRequest{
 //
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ReactivateConfigurationRequest.
}
op, err := c.ReactivateConfiguration(ctx, req)
if err != nil {
 // TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleClient_UpdateConfiguration() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.UpdateConfigurationRequest{
 // TODO: Fill request
 struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#UpdateConfigurationRequest.
 }
 op, err := c.UpdateConfiguration(ctx, req)
 if err != nil {

```

```

// TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleClient_GetLocation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &locationpb.GetLocationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/google.golang.org/genproto/googleapis/cloud/location#GetLocationRequest.
 }
 resp,
 err := c.GetLocation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleClient_ListLocations() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

defer c.Close()

req := &locationpb.ListLocationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/google.golang.org/genproto/googleapis/cloud/location#ListLocationsRequest.
}
it := c.ListLocations(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if
err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*locationpb.ListLocationsResponse)
}
}

func ExampleClient_CancelOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.CancelOperationRequest{
 // TODO: Fill request struct
 fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#CancelOperationRequest.
 }
 err = c.CancelOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.

```

```

}
}

func ExampleClient_DeleteOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.DeleteOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#DeleteOperationRequest.
 }
 err = c.DeleteOperation(ctx, req)
 if err != nil {
 //
 // TODO: Handle error.
 }
}

func ExampleClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }
 resp, err := c.GetOperation(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

// TODO: Use resp.
_ = resp
}

func ExampleClient_ListOperations() {
 ctx := context.Background()
 // This snippet has been automatically generated and
 // should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
 }
 it := c.ListOperations(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next().
 Not safe for concurrent access.
 _ = it.Response.(*longrunningpb.ListOperationsResponse)
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//

```

```
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}
```

```

defer c.Close()

req := &computepb.GetLicenseCodeRequest{
 // TODO:
 Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseCodeRequest.
}
resp, err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseCodeRequest.
 }
 resp,
 err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0

```



```

//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package licensemanager

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 licensemanagerpb "cloud.google.com/go/licensemanager/apiv1/licensemanagerpb"
 "cloud.google.com/go/longrunning"
 lroauto "cloud.google.com/go/longrunning/autogen"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport
 "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 locationpb "google.golang.org/genproto/googleapis/cloud/location"
 "google.golang.org/grpc"
 "google.golang.org/grpc/codes"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newClientHook clientHook

// CallOptions contains the retry settings for each method of Client.
type CallOptions struct {
 ListConfigurations []gax.CallOption
 GetConfiguration []gax.CallOption
 CreateConfiguration []gax.CallOption
 UpdateConfiguration []gax.CallOption
 DeleteConfiguration []gax.CallOption

```

```

ListInstances []gax.CallOption
GetInstance []gax.CallOption
DeactivateConfiguration []gax.CallOption
ReactivateConfiguration
 []gax.CallOption
QueryConfigurationLicenseUsage []gax.CallOption
AggregateUsage []gax.CallOption
ListProducts []gax.CallOption
GetProduct []gax.CallOption
GetLocation []gax.CallOption
ListLocations []gax.CallOption
CancelOperation []gax.CallOption
DeleteOperation []gax.CallOption
GetOperation []gax.CallOption
ListOperations []gax.CallOption
}

```

```

func defaultGRPCClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("licensemanager.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("licensemanager.UNIVERSE_DOMAIN:443"),
 internaloption.WithDefaultMTLSEndpoint("licensemanager.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://licensemanager.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
}
}

```

```

func
defaultCallOptions() *CallOptions {
return &CallOptions{
 ListConfigurations: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
 },
 },
 },
 GetConfiguration: []gax.CallOption{

```

```

gax.WithTimeout(60000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
}),
},
CreateConfiguration: []gax.CallOption{
 gax.WithTimeout(60000
* time.Millisecond),
},
UpdateConfiguration: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
},
DeleteConfiguration: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
},
ListInstances: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
GetInstance: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
DeactivateConfiguration: []gax.CallOption{},
ReactivateConfiguration: []gax.CallOption{},
QueryConfigurationLicenseUsage:

```

```

[]gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
AggregateUsage: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
},
ListProducts: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
GetProduct: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max:
10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
GetLocation: []gax.CallOption{},
ListLocations: []gax.CallOption{},
CancelOperation: []gax.CallOption{},
DeleteOperation: []gax.CallOption{},
GetOperation: []gax.CallOption{},
ListOperations: []gax.CallOption{},
}
}

```

```

func defaultRESTCallOptions() *CallOptions {
return &CallOptions{
ListConfigurations: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 10000 * time.Millisecond,
Multiplier: 1.30,
}),
http.StatusServiceUnavailable)
}),
},
GetConfiguration: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 10000 * time.Millisecond,
Multiplier: 1.30,
}),
http.StatusServiceUnavailable)
}),
},
CreateConfiguration:
[]gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
},
UpdateConfiguration: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
},
DeleteConfiguration: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
},
ListInstances: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
Initial: 1000 * time.Millisecond,
Max: 10000 * time.Millisecond,
Multiplier: 1.30,
}),
http.StatusServiceUnavailable)
}),
},
GetInstance: []gax.CallOption{
gax.WithTimeout(60000 * time.Millisecond),

```

```

gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
}),
 http.StatusServiceUnavailable)
}),
},
DeactivateConfiguration: []gax.CallOption{},
ReactivateConfiguration:
[]gax.CallOption{},
QueryConfigurationLicenseUsage: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
}),
 http.StatusServiceUnavailable)
}),
},
AggregateUsage: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
},
ListProducts: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
}),
 http.StatusServiceUnavailable)
}),
},
GetProduct: []gax.CallOption{
 gax.WithTimeout(60000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max:
10000 * time.Millisecond,
 Multiplier: 1.30,
}),
 http.StatusServiceUnavailable)
}),
},

```

```

 },
 GetLocation: []gax.CallOption{},
 ListLocations: []gax.CallOption{},
 CancelOperation: []gax.CallOption{},
 DeleteOperation: []gax.CallOption{},
 GetOperation: []gax.CallOption{},
 ListOperations: []gax.CallOption{},
}
}

// internalClient is an interface that defines the methods available from License Manager API.
type internalClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 ListConfigurations(context.Context, *licensemanagerpb.ListConfigurationsRequest, ...gax.CallOption)
 *ConfigurationIterator
 GetConfiguration(context.Context, *licensemanagerpb.GetConfigurationRequest, ...gax.CallOption)
 (*licensemanagerpb.Configuration, error)
 CreateConfiguration(context.Context, *licensemanagerpb.CreateConfigurationRequest, ...gax.CallOption)
 (*CreateConfigurationOperation, error)
 CreateConfigurationOperation(name string) *CreateConfigurationOperation
 UpdateConfiguration(context.Context,
 *licensemanagerpb.UpdateConfigurationRequest, ...gax.CallOption) (*UpdateConfigurationOperation, error)
 UpdateConfigurationOperation(name string) *UpdateConfigurationOperation
 DeleteConfiguration(context.Context, *licensemanagerpb.DeleteConfigurationRequest, ...gax.CallOption)
 (*DeleteConfigurationOperation, error)
 DeleteConfigurationOperation(name string) *DeleteConfigurationOperation
 ListInstances(context.Context, *licensemanagerpb.ListInstancesRequest, ...gax.CallOption) *InstanceIterator
 GetInstance(context.Context, *licensemanagerpb.GetInstanceRequest, ...gax.CallOption)
 (*licensemanagerpb.Instance, error)
 DeactivateConfiguration(context.Context, *licensemanagerpb.DeactivateConfigurationRequest, ...gax.CallOption)
 (*DeactivateConfigurationOperation, error)
 DeactivateConfigurationOperation(name string) *DeactivateConfigurationOperation
 ReactivateConfiguration(context.Context, *licensemanagerpb.ReactivateConfigurationRequest,
 ...gax.CallOption) (*ReactivateConfigurationOperation, error)
 ReactivateConfigurationOperation(name string) *ReactivateConfigurationOperation
 QueryConfigurationLicenseUsage(context.Context, *licensemanagerpb.QueryConfigurationLicenseUsageRequest,
 ...gax.CallOption) (*licensemanagerpb.QueryConfigurationLicenseUsageResponse, error)
 AggregateUsage(context.Context, *licensemanagerpb.AggregateUsageRequest, ...gax.CallOption) *UsageIterator
 ListProducts(context.Context, *licensemanagerpb.ListProductsRequest, ...gax.CallOption) *ProductIterator
 GetProduct(context.Context, *licensemanagerpb.GetProductRequest, ...gax.CallOption)
 (*licensemanagerpb.Product, error)
 GetLocation(context.Context, *locationpb.GetLocationRequest, ...gax.CallOption) (*locationpb.Location, error)
 ListLocations(context.Context, *locationpb.ListLocationsRequest, ...gax.CallOption) *LocationIterator
 CancelOperation(context.Context, *longrunningpb.CancelOperationRequest, ...gax.CallOption) error
 DeleteOperation(context.Context,

```

```

*longrunningpb.DeleteOperationRequest, ...gax.CallOption) error
GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
ListOperations(context.Context, *longrunningpb.ListOperationsRequest, ...gax.CallOption) *OperationIterator
}

// Client is a client for interacting with License Manager API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// Service describing handlers for resources
type Client struct {
 // The internal transport-dependent client.
 internalClient internalClient

 // The call options for this service.
 CallOptions *CallOptions

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient *lroauto.OperationsClient
}

// Wrapper methods routed to the internal client.

// Close
// closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *Client) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *Client) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *Client) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

```



```

// ListConfigurations lists Configurations in a given project and location.
func (c *Client) ListConfigurations(ctx context.Context, req *licensemanagerpb.ListConfigurationsRequest, opts
...gax.CallOption) *ConfigurationIterator {
 return c.internalClient.ListConfigurations(ctx,
 req, opts...)
}

// GetConfiguration gets details of a single Configuration.
func (c *Client) GetConfiguration(ctx context.Context, req *licensemanagerpb.GetConfigurationRequest, opts
...gax.CallOption) (*licensemanagerpb.Configuration, error) {
 return c.internalClient.GetConfiguration(ctx, req, opts...)
}

// CreateConfiguration creates a new Configuration in a given project and location.
func (c *Client) CreateConfiguration(ctx context.Context, req *licensemanagerpb.CreateConfigurationRequest, opts
...gax.CallOption) (*CreateConfigurationOperation, error) {
 return c.internalClient.CreateConfiguration(ctx, req, opts...)
}

// CreateConfigurationOperation returns a new CreateConfigurationOperation from a given name.
// The name must be that of a previously created CreateConfigurationOperation, possibly from a different process.
func (c *Client) CreateConfigurationOperation(name string) *CreateConfigurationOperation {
 return c.internalClient.CreateConfigurationOperation(name)
}

//
// UpdateConfiguration updates the parameters of a single Configuration.
func (c *Client) UpdateConfiguration(ctx context.Context, req *licensemanagerpb.UpdateConfigurationRequest,
opts ...gax.CallOption) (*UpdateConfigurationOperation, error) {
 return c.internalClient.UpdateConfiguration(ctx, req, opts...)
}

// UpdateConfigurationOperation returns a new UpdateConfigurationOperation from a given name.
// The name must be that of a previously created UpdateConfigurationOperation, possibly from a different process.
func (c *Client) UpdateConfigurationOperation(name string) *UpdateConfigurationOperation {
 return c.internalClient.UpdateConfigurationOperation(name)
}

// DeleteConfiguration deletes a single Configuration.
func (c *Client) DeleteConfiguration(ctx context.Context, req *licensemanagerpb.DeleteConfigurationRequest, opts
...gax.CallOption) (*DeleteConfigurationOperation, error) {
 return c.internalClient.DeleteConfiguration(ctx, req, opts...)
}

// DeleteConfigurationOperation returns
// a new DeleteConfigurationOperation from a given name.
// The name must be that of a previously created DeleteConfigurationOperation, possibly from a different process.
func (c *Client) DeleteConfigurationOperation(name string) *DeleteConfigurationOperation {

```

```

return c.internalClient.DeleteConfigurationOperation(name)
}

// ListInstances lists Instances in a given project and location.
func (c *Client) ListInstances(ctx context.Context, req *licensemanagerpb.ListInstancesRequest, opts
...gax.CallOption) *InstanceIterator {
return c.internalClient.ListInstances(ctx, req, opts...)
}

// GetInstance gets details of a single Instance.
func (c *Client) GetInstance(ctx context.Context, req *licensemanagerpb.GetInstanceRequest, opts
...gax.CallOption) (*licensemanagerpb.Instance, error) {
return c.internalClient.GetInstance(ctx, req, opts...)
}

// DeactivateConfiguration deactivates the given configuration.
func (c *Client) DeactivateConfiguration(ctx context.Context, req
*licensemanagerpb.DeactivateConfigurationRequest,
opts ...gax.CallOption) (*DeactivateConfigurationOperation, error) {
return c.internalClient.DeactivateConfiguration(ctx, req, opts...)
}

// DeactivateConfigurationOperation returns a new DeactivateConfigurationOperation from a given name.
// The name must be that of a previously created DeactivateConfigurationOperation, possibly from a different
process.
func (c *Client) DeactivateConfigurationOperation(name string) *DeactivateConfigurationOperation {
return c.internalClient.DeactivateConfigurationOperation(name)
}

// ReactivateConfiguration reactivates the given configuration.
func (c *Client) ReactivateConfiguration(ctx context.Context, req
*licensemanagerpb.ReactivateConfigurationRequest, opts ...gax.CallOption) (*ReactivateConfigurationOperation,
error) {
return c.internalClient.ReactivateConfiguration(ctx, req, opts...)
}

// ReactivateConfigurationOperation returns a new ReactivateConfigurationOperation from a given name.
// The name must
be that of a previously created ReactivateConfigurationOperation, possibly from a different process.
func (c *Client) ReactivateConfigurationOperation(name string) *ReactivateConfigurationOperation {
return c.internalClient.ReactivateConfigurationOperation(name)
}

// QueryConfigurationLicenseUsage license Usage information for a Configuration.
func (c *Client) QueryConfigurationLicenseUsage(ctx context.Context, req
*licensemanagerpb.QueryConfigurationLicenseUsageRequest, opts ...gax.CallOption)
(*licensemanagerpb.QueryConfigurationLicenseUsageResponse, error) {
return c.internalClient.QueryConfigurationLicenseUsage(ctx, req, opts...)
}

```

```

}

// AggregateUsage aggregates Usage per Instance for a Configuration.
func (c *Client) AggregateUsage(ctx context.Context, req *licensemanagerpb.AggregateUsageRequest, opts
...gax.CallOption) *UsageIterator {
 return c.internalClient.AggregateUsage(ctx, req, opts...)
}

// ListProducts lists Products in a given project and location.
func (c *Client)
 ListProducts(ctx context.Context, req *licensemanagerpb.ListProductsRequest, opts ...gax.CallOption)
*ProductIterator {
 return c.internalClient.ListProducts(ctx, req, opts...)
}

// GetProduct gets details of a single Product.
func (c *Client) GetProduct(ctx context.Context, req *licensemanagerpb.GetProductRequest, opts ...gax.CallOption)
(*licensemanagerpb.Product, error) {
 return c.internalClient.GetProduct(ctx, req, opts...)
}

// GetLocation gets information about a location.
func (c *Client) GetLocation(ctx context.Context, req *locationpb.GetLocationRequest, opts ...gax.CallOption)
(*locationpb.Location, error) {
 return c.internalClient.GetLocation(ctx, req, opts...)
}

// ListLocations lists information about the supported locations for this service.
func (c *Client) ListLocations(ctx context.Context, req *locationpb.ListLocationsRequest, opts ...gax.CallOption)
*LocationIterator {
 return c.internalClient.ListLocations(ctx, req, opts...)
}

// CancelOperation is a utility
method from google.longrunning.Operations.
func (c *Client) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest, opts
...gax.CallOption) error {
 return c.internalClient.CancelOperation(ctx, req, opts...)
}

// DeleteOperation is a utility method from google.longrunning.Operations.
func (c *Client) DeleteOperation(ctx context.Context, req *longrunningpb.DeleteOperationRequest, opts
...gax.CallOption) error {
 return c.internalClient.DeleteOperation(ctx, req, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *Client) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts

```

```

...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *Client) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest, opts
...gax.CallOption) *OperationIterator {
 return
 c.internalClient.ListOperations(ctx, req, opts...)
}

// gRPCClient is a client for interacting with License Manager API over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type gRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing Client
 CallOptions **CallOptions

 // The gRPC API client.
 client licensemanagerpb.LicenseManagerClient

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 operationsClient longrunningpb.OperationsClient

 locationsClient locationpb.LocationsClient

 // The x-goog-* metadata to be sent with each request.
 xGoogHeaders []string

 logger *slog.Logger
}

// NewClient creates a new license
manager client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service describing handlers for resources
func NewClient(ctx context.Context, opts ...option.ClientOption) (*Client, error) {
 clientOpts := defaultGRPCClientOptions()
 if newClientHook != nil {

```

```

hookOpts, err := newClientHook(ctx, clientHookParams{})
if err != nil {
 return nil, err
}
clientOpts = append(clientOpts, hookOpts...)
}

connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
if err != nil {
 return nil, err
}
client := Client{CallOptions: defaultCallOptions()}

c := &gRPCClient{
 connPool: connPool,
 client: licensemanagerpb.NewLicenseManagerClient(connPool),
 CallOptions: &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
 locationsClient: locationpb.NewLocationsClient(connPool),
}
c.setGoogleClientInfo()

client.internalClient
= c

client.LROClient, err = lroauto.NewOperationsClient(ctx, gtransport.WithConnPool(connPool))
if err != nil {
 // This error "should not happen", since we are just reusing old connection pool
 // and never actually need to dial.
 // If this does happen, we could leak connp. However, we cannot close conn:
 // If the user invoked the constructor with option.WithGRPCConn,
 // we would close a connection that's still in use.
 // TODO: investigate error conditions.
 return nil, err
}
c.LROClient = &client.LROClient
return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *gRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

```

```

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written
clients.
func (c *gRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc", grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *gRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type restClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 // The x-goog-*
 headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing Client
 CallOptions **CallOptions

 logger *slog.Logger
}

// NewRESTClient creates a new license manager rest client.
//
// Service describing handlers for resources
func NewRESTClient(ctx context.Context, opts ...option.ClientOption) (*Client, error) {
 clientOpts := append(defaultRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)

```

```

if err != nil {
 return nil, err
}

callOpts := defaultRESTCallOptions()
c := &restClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

lroOpts := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opClient, err := lroauto.NewOperationsRESTClient(ctx, lroOpts...)
if err != nil {
 return nil, err
}
c.LROClient = &opClient

return
&Client{internalClient: c, CallOptions: callOpts}, nil
}

func defaultRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://licensemanager.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://licensemanager.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://licensemanager.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://licensemanager.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *restClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(),
 "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

```

```

}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *restClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *restClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *gRPCClient) ListConfigurations(ctx context.Context, req *licensemanagerpb.ListConfigurationsRequest,
 opts ...gax.CallOption) *ConfigurationIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).ListConfigurations[0:len((*c.CallOptions).ListConfigurations):len((*c.CallOptions).ListCo
 nfigurations)],
 opts...)
 it := &ConfigurationIterator{}
 req = proto.Clone(req).(*licensemanagerpb.ListConfigurationsRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Configuration, string, error) {
 resp := &licensemanagerpb.ListConfigurationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.ListConfigurations, req, settings.GRPC, c.logger, "ListConfigurations")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }
 }
}

```



```

it.Response = resp
return resp.GetConfigurations(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize,
pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *gRPCClient) GetConfiguration(ctx context.Context, req *licensemanagerpb.GetConfigurationRequest, opts
...gax.CallOption) (*licensemanagerpb.Configuration, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetConfiguration[0:len((*c.CallOptions).GetConfiguration):len((*c.CallOptions).GetConf
guration)], opts...)
var resp *licensemanagerpb.Configuration
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx,
c.client.GetConfiguration, req, settings.GRPC, c.logger, "GetConfiguration")
return err
}, opts...)
if err != nil {
return nil, err
}
return resp, nil
}

func (c *gRPCClient) CreateConfiguration(ctx context.Context, req
*licensemanagerpb.CreateConfigurationRequest, opts ...gax.CallOption) (*CreateConfigurationOperation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)

```

```

opts =
append((*c.CallOptions).CreateConfiguration[0:len((*c.CallOptions).CreateConfiguration):len((*c.CallOptions).CreateConfiguration)], opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.CreateConfiguration, req, settings.GRPC, c.logger, "CreateConfiguration")
 return err
}, opts...)
if err != nil {
 return
 nil, err
}
return &CreateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 }, nil
}

```

```

func (c *gRPCClient) UpdateConfiguration(ctx context.Context, req
*licensemanagerpb.UpdateConfigurationRequest, opts ...gax.CallOption) (*UpdateConfigurationOperation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "configuration.name",
 url.QueryEscape(req.GetConfiguration().GetName()))}

```

```

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).UpdateConfiguration[0:len((*c.CallOptions).UpdateConfiguration):len((*c.CallOptions).UpdateConfiguration)], opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.UpdateConfiguration, req, settings.GRPC, c.logger, "UpdateConfiguration")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return
&UpdateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 }, nil
}

```

```

func (c *gRPCClient) DeleteConfiguration(ctx context.Context, req
*licensemanagerpb.DeleteConfigurationRequest, opts ...gax.CallOption) (*DeleteConfigurationOperation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

 hds = append(c.xGoogHeaders, hds...)

```

```

ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).DeleteConfiguration[0:len((*c.CallOptions).DeleteConfiguration):len((*c.CallOptions).DeleteConfiguration)], opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.DeleteConfiguration, req, settings.GRPC, c.logger, "DeleteConfiguration")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return &DeleteConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient,
resp),
}, nil
}

func (c *gRPCClient) ListInstances(ctx context.Context, req *licensemanagerpb.ListInstancesRequest, opts
...gax.CallOption) *InstanceIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).ListInstances[0:len((*c.CallOptions).ListInstances):len((*c.CallOptions).ListInstances)],
opts...)
it := &InstanceIterator{
req = proto.Clone(req).(*licensemanagerpb.ListInstancesRequest)
it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Instance, string, error) {
 resp := &licensemanagerpb.ListInstancesResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx,
func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.ListInstances, req, settings.GRPC, c.logger, "ListInstances")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }
}

```

```

 }

 it.Response = resp
 return resp.GetInstances(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *gRPCClient) GetInstance(ctx context.Context, req *licensemanagerpb.GetInstanceRequest, opts
...gax.CallOption) (*licensemanagerpb.Instance, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds
 = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetInstance[0:len((*c.CallOptions).GetInstance):len((*c.CallOptions).GetInstance)],
 opts...)
 var resp *licensemanagerpb.Instance
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.GetInstance, req, settings.GRPC, c.logger, "GetInstance")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *gRPCClient) DeactivateConfiguration(ctx context.Context, req
*licensemanagerpb.DeactivateConfigurationRequest, opts ...gax.CallOption) (*DeactivateConfigurationOperation,
error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).DeactivateConfiguration[0:len((*c.CallOptions).DeactivateConfiguration):len((*c.CallOptions).DeactivateConfiguration)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.DeactivateConfiguration, req, settings.GRPC, c.logger,
"DeactivateConfiguration")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return &DeactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
}, nil
}

```

```

func (c *gRPCClient) ReactivateConfiguration(ctx context.Context, req
*licensemanagerpb.ReactivateConfigurationRequest, opts ...gax.CallOption) (*ReactivateConfigurationOperation,
error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).ReactivateConfiguration[0:len((*c.CallOptions).ReactivateConfiguration):len((*c.CallOptions).ReactivateConfiguration)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.ReactivateConfiguration, req, settings.GRPC, c.logger,
"ReactivateConfiguration")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return &ReactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
}, nil
}

```

```

func (c *gRPCClient) QueryConfigurationLicenseUsage(ctx context.Context, req

```

```

*licensemanagerpb.QueryConfigurationLicenseUsageRequest, opts ...gax.CallOption)
(*licensemanagerpb.QueryConfigurationLicenseUsageResponse, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).QueryConfigurationLicenseUsage[0:len((*c.CallOptions).QueryConfigurationLicenseUsage):len((*c.CallOptions).QueryConfigurationLicenseUsage)],
 opts...)
 var resp *licensemanagerpb.QueryConfigurationLicenseUsageResponse
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.QueryConfigurationLicenseUsage, req, settings.GRPC, c.logger,
 "QueryConfigurationLicenseUsage")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```

```

func (c *gRPCClient) AggregateUsage(ctx context.Context, req *licensemanagerpb.AggregateUsageRequest, opts
...gax.CallOption) *UsageIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).AggregateUsage[0:len((*c.CallOptions).AggregateUsage):len((*c.CallOptions).Aggregate
 Usage)],
 opts...)
 it := &UsageIterator{}
 req = proto.Clone(req).(*licensemanagerpb.AggregateUsageRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Usage, string, error) {
 resp := &licensemanagerpb.AggregateUsageResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.AggregateUsage, req, settings.GRPC, c.logger, "AggregateUsage")

```

```

 return err
}, opts...)
if err != nil {
 return nil, "", err
}

it.Response = resp
return resp.GetUsages(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items,
items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *gRPCClient) ListProducts(ctx context.Context, req *licensemanagerpb.ListProductsRequest, opts
...gax.CallOption) *ProductIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).ListProducts[0:len((*c.CallOptions).ListProducts):len((*c.CallOptions).ListProducts)],
opts...)
 it := &ProductIterator{}
 req = proto.Clone(req).(*licensemanagerpb.ListProductsRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Product, string, error) {
 resp := &licensemanagerpb.ListProductsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if
 pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 }
}

```

```

err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.ListProducts, req, settings.GRPC, c.logger, "ListProducts")
 return err
}, opts...)
if err != nil {
 return nil, "", err
}

it.Response = resp
return resp.GetProducts(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *gRPCClient) GetProduct(ctx context.Context, req *licensemanagerpb.GetProductRequest,
 opts ...gax.CallOption) (*licensemanagerpb.Product, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts = append((*c.CallOptions).GetProduct[0:len((*c.CallOptions).GetProduct):len((*c.CallOptions).GetProduct)],
 opts...)
 var resp *licensemanagerpb.Product
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.client.GetProduct, req, settings.GRPC, c.logger, "GetProduct")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

```



```

func (c *gRPCClient) GetLocation(ctx context.Context, req *locationpb.GetLocationRequest, opts
...gax.CallOption) (*locationpb.Location, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders,
 hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).GetLocation[0:len((*c.CallOptions).GetLocation):len((*c.CallOptions).GetLocation)],
 opts...)
 var resp *locationpb.Location
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.locationsClient.GetLocation, req, settings.GRPC, c.logger, "GetLocation")
 return err
 }, opts...)
 if err != nil {
 return nil, err
 }
 return resp, nil
}

func (c *gRPCClient) ListLocations(ctx context.Context, req *locationpb.ListLocationsRequest, opts
...gax.CallOption) *LocationIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).ListLocations[0:len((*c.CallOptions).ListLocations):len((*c.CallOptions).ListLocations)],
 opts...)
 it := &LocationIterator{}
 req
 = proto.Clone(req).(*locationpb.ListLocationsRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*locationpb.Location, string, error) {
 resp := &locationpb.ListLocationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.locationsClient.ListLocations, req, settings.GRPC, c.logger, "ListLocations")
 return err

```

```

 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetLocations(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken,
 nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *gRPCClient) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest, opts
...gax.CallOption) error {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
 append((*c.CallOptions).CancelOperation[0:len((*c.CallOptions).CancelOperation):len((*c.CallOptions).CancelOp
eration)], opts...)
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 _, err = executeRPC(ctx, c.operationsClient.CancelOperation, req, settings.GRPC, c.logger, "CancelOperation")
 return err
 }, opts...)
 return err
}

func (c *gRPCClient) DeleteOperation(ctx context.Context, req *longrunningpb.DeleteOperationRequest,
opts ...gax.CallOption) error {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)

```

```

opts =
append((*c.CallOptions).DeleteOperation[0:len((*c.CallOptions).DeleteOperation):len((*c.CallOptions).DeleteOperation)], opts...)
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 _, err = executeRPC(ctx, c.operationsClient.DeleteOperation, req, settings.GRPC, c.logger, "DeleteOperation")
 return err
}, opts...)
return err
}

```

```

func (c *gRPCClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts
...gax.CallOption) (*longrunningpb.Operation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx,
hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

```

```

func (c *gRPCClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest, opts
...gax.CallOption) *OperationIterator {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).ListOperations[0:len((*c.CallOptions).ListOperations):len((*c.CallOptions).ListOperations)], opts...)
it := &OperationIterator{ }
req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
it.InternalFetch
= func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{ }
 if pageToken != "" {

```

```

 req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
} else if pageSize != 0 {
 req.PageSize = int32(pageSize)
}
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.ListOperations, req, settings.GRPC, c.logger, "ListOperations")
 return err
}, opts...)
if err != nil {
 return nil, "", err
}

it.Response = resp
return resp.GetOperations(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo,
it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// ListConfigurations lists Configurations in a given project and location.
func (c *restClient) ListConfigurations(ctx context.Context, req *licensemanagerpb.ListConfigurationsRequest, opts
...gax.CallOption) *ConfigurationIterator {
 it := &ConfigurationIterator{}
 req = proto.Clone(req).(*licensemanagerpb.ListConfigurationsRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Configuration, string, error) {
 resp := &licensemanagerpb.ListConfigurationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {

```

```

 req.PageSize = math.MaxInt32
} else if pageSize != 0 {
 req.PageSize = int32(pageSize)
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return
 nil, "", err
}
baseUrl.Path += fmt.Sprintf("/v1/%v/configurations", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req.GetOrderBy() != "" {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil
{
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListConfigurations")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetConfigurations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetConfiguration gets details of a single Configuration.
func (c *restClient) GetConfiguration(ctx context.Context, req *licensemanagerpb.GetConfigurationRequest,
 opts ...gax.CallOption) (*licensemanagerpb.Configuration, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =

```

```

append((*c.CallOptions).GetConfiguration[0:len((*c.CallOptions).GetConfiguration):len((*c.CallOptions).GetConfiguration)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &licensemanagerpb.Configuration{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
 = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetConfiguration")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// CreateConfiguration creates a new Configuration in a given project and location.
func (c *restClient) CreateConfiguration(ctx context.Context, req *licensemanagerpb.CreateConfigurationRequest,
opts ...gax.CallOption) (*CreateConfigurationOperation, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetConfiguration()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path

```

```

+= fmt.Sprintf("/v1/%v/configurations", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
params.Add("configurationId", fmt.Sprintf("%v", req.GetConfigurationId()))
if req.GetRequestId() != "" {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err !=
nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "CreateConfiguration")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &CreateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),

```



```

 pollPath: override,
 }, nil
}

// UpdateConfiguration updates the parameters of a single Configuration.
func (c *restClient) UpdateConfiguration(ctx context.Context, req *licensemanagerpb.UpdateConfigurationRequest,
opts ...gax.CallOption) (*UpdateConfigurationOperation, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 body := req.GetConfiguration()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
 err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetConfiguration().GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetRequestId() != "" {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
 if req.GetUpdateMask() != nil {
 field, err := protojson.Marshal(req.GetUpdateMask())
 if err != nil {
 return nil, err
 }
 params.Add("updateMask", string(field[1:len(field)-1]))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "configuration.name",
url.QueryEscape(req.GetConfiguration().GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context,
settings gax.CallSettings) error {
 if settings.Path != "" {

```

```

 baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
if err != nil {
 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "UpdateConfiguration")
if err != nil {
 return err
}
if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &UpdateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// DeleteConfiguration deletes a single Configuration.
func (c *restClient) DeleteConfiguration(ctx context.Context, req *licensemanagerpb.DeleteConfigurationRequest,
opts ...gax.CallOption) (*DeleteConfigurationOperation, error) {
 baseUrl,
 err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetRequestId() != "" {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

```

```

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq
 = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "DeleteConfiguration")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &DeleteConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// ListInstances lists Instances in a given project and location.
func (c *restClient) ListInstances(ctx context.Context, req *licensemanagerpb.ListInstancesRequest, opts
...gax.CallOption) *InstanceIterator {
 it := &InstanceIterator{}
 req = proto.Clone(req).(*licensemanagerpb.ListInstancesRequest)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*licensemanagerpb.Instance, string, error) {

```

```

resp
:= &licensemanagerpb.ListInstancesResponse{}
if pageToken != "" {
 req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
} else if pageSize != 0 {
 req.PageSize = int32(pageSize)
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/v1/%v/instances", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req.GetOrderBy() != "" {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
}
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers
:= gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListInstances")

```

```

 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetInstances(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen,
it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetInstance gets details of a single Instance.
func (c *restClient) GetInstance(ctx context.Context, req *licensemanagerpb.GetInstanceRequest, opts
...gax.CallOption) (*licensemanagerpb.Instance, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.

```

```

hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetInstance[0:len((*c.CallOptions).GetInstance):len((*c.CallOptions).GetInstance)],
opts...)
unm :=
protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &licensemanagerpb.Instance{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
if settings.Path != "" {
baseUrl.Path = settings.Path
}
httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
if err != nil {
return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetInstance")
if err != nil {
return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
return err
}

return nil
}, opts...)
if e != nil {
return nil, e
}
return resp, nil
}

// DeactivateConfiguration deactivates the given configuration.
func (c *restClient) DeactivateConfiguration(ctx context.Context, req
*licensemanagerpb.DeactivateConfigurationRequest, opts ...gax.CallOption) (*DeactivateConfigurationOperation,
error) {
m := protojson.MarshalOptions{AllowPartial: true,
UseEnumNumbers: true}
jsonReq, err := m.Marshal(req)
if err != nil {
return nil, err

```

```

}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v:deactivate", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 {
 return err
 }
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "DeactivateConfiguration")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())

```

```

return &DeactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// ReactivateConfiguration reactivates the given configuration.
func (c *restClient) ReactivateConfiguration(ctx context.Context, req
*licensemanagerpb.ReactivateConfigurationRequest, opts ...gax.CallOption) (*ReactivateConfigurationOperation,
error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err !=
 nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:reactivate", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &longrunningpb.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq,

```



```

c.logger, jsonReq, "ReactivateConfiguration")
if err != nil {
 return err
}
if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &ReactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// QueryConfigurationLicenseUsage license Usage information for a Configuration.
func (c *restClient) QueryConfigurationLicenseUsage(ctx context.Context, req
*licensemanagerpb.QueryConfigurationLicenseUsageRequest, opts ...gax.CallOption)
(*licensemanagerpb.QueryConfigurationLicenseUsageResponse, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:queryLicenseUsage", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetEndTime() != nil {
 field, err := protojson.Marshal(req.GetEndTime())
 if
err != nil {
 return nil, err
 }
 params.Add("endTime", string(field[1:len(field)-1]))
 }
 if req.GetStartTime() != nil {
 field, err := protojson.Marshal(req.GetStartTime())
 if err != nil {
 return nil, err
 }
 params.Add("startTime", string(field[1:len(field)-1]))
 }
}

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).QueryConfigurationLicenseUsage[0:len((*c.CallOptions).QueryConfigurationLicenseUsage):len((*c.CallOptions).QueryConfigurationLicenseUsage)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &licensemanagerpb.QueryConfigurationLicenseUsageResponse{}
e := gax.Invoke(ctx, func(ctx
context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "QueryConfigurationLicenseUsage")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// AggregateUsage aggregates Usage per Instance for a Configuration.
func (c *restClient) AggregateUsage(ctx context.Context, req *licensemanagerpb.AggregateUsageRequest, opts
...gax.CallOption) *UsageIterator {
 it := &UsageIterator{}
 req = proto.Clone(req).(*licensemanagerpb.AggregateUsageRequest)

```

```

unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
it.InternalFetch = func(pageSize int, pageToken
string) ([]*licensemanagerpb.Usage, string, error) {
 resp := &licensemanagerpb.AggregateUsageResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:aggregateUsage", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetEndTime() != nil {
 field, err := protojson.Marshal(req.GetEndTime())
 if err != nil {
 return nil, "", err
 }
 params.Add("endTime", string(field[1:len(field)-1]))
 }
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req.GetOrderBy() != "" {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if
 req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }
 if req.GetStartTime() != nil {
 field, err := protojson.Marshal(req.GetStartTime())
 if err != nil {
 return nil, "", err
 }
 params.Add("startTime", string(field[1:len(field)-1]))
 }
}

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "AggregateUsage")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return
nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetUsages(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

```

```

// ListProducts lists Products in a given project and location.
func (c *restClient) ListProducts(ctx context.Context, req *licensemanagerpb.ListProductsRequest, opts
...gax.CallOption) *ProductIterator {
 it := &ProductIterator{}
 req = proto.Clone(req).(*licensemanagerpb.ListProductsRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken
string) ([]*licensemanagerpb.Product, string, error) {
 resp := &licensemanagerpb.ListProductsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/products", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req.GetOrderBy() != "" {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds
 := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }

```

```

 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListProducts")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetProducts(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo,
it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// GetProduct gets details of a single Product.
func (c *restClient) GetProduct(ctx context.Context, req *licensemanagerpb.GetProductRequest, opts
...gax.CallOption) (*licensemanagerpb.Product, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }

```

```

baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).GetProduct[0:len((*c.CallOptions).GetProduct):len((*c.CallOptions).GetProduct)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &licensemanagerpb.Product{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetProduct")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetLocation gets information about a location.
func (c *restClient) GetLocation(ctx context.Context, req *locationpb.GetLocationRequest, opts ...gax.CallOption)
(*locationpb.Location, error) {

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return
 nil, err
}
baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetLocation[0:len((*c.CallOptions).GetLocation):len((*c.CallOptions).GetLocation)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &locationpb.Location{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header
= headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetLocation")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e

```



```

}
return resp, nil
}

// ListLocations lists information about the supported locations for this service.
func (c *restClient) ListLocations(ctx context.Context, req *locationpb.ListLocationsRequest, opts
...gax.CallOption) *LocationIterator {
 it := &LocationIterator{}
 req = proto.Clone(req).(*locationpb.ListLocationsRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*locationpb.Location, string, error) {
 resp := &locationpb.ListLocationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl,
 err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/locations", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }

```

```

httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
if err != nil {
 return err
}
httpReq.Header
= headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListLocations")
if err != nil {
 return err
}
if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetLocations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *restClient) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest, opts
...gax.CallOption)
error {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return err
 }

```

```

}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return err
}
baseUrl.Path += fmt.Sprintf("/v1/%v:cancel", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
return gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header
 = headers

 _, err = executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "CancelOperation")
 return err
}, opts...)
}

// DeleteOperation is a utility method from google.longrunning.Operations.
func (c *restClient) DeleteOperation(ctx context.Context, req *longrunningpb.DeleteOperationRequest, opts
...gax.CallOption) error {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
return gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path !=
"" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 _, err = executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "DeleteOperation")
 return err
}, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *restClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts
...gax.CallOption) (*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

 baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],

```

```

opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// ListOperations is a utility method from google.longrunning.Operations.
func
(c *restClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest, opts
...gax.CallOption) *OperationIterator {
 it := &OperationIterator{}
 req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
 unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 }
}

```

```

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/v1/%v/operations", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
}
if req.GetPageSize() != 0 {
 params.Add("pageSize",
fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListOperations")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp

```

```

return resp.GetOperations(), resp.GetNextPageToken(),
nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// CreateConfigurationOperation returns a new CreateConfigurationOperation from a given name.
// The name must be that of a previously created CreateConfigurationOperation, possibly from a different process.
func (c *gRPCClient) CreateConfigurationOperation(name string) *CreateConfigurationOperation {
return &CreateConfigurationOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
}
}

// CreateConfigurationOperation returns a new CreateConfigurationOperation from a given name.
// The name must
be that of a previously created CreateConfigurationOperation, possibly from a different process.
func (c *restClient) CreateConfigurationOperation(name string) *CreateConfigurationOperation {
override := fmt.Sprintf("/v1/%s", name)
return &CreateConfigurationOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
pollPath: override,
}
}

// DeactivateConfigurationOperation returns a new DeactivateConfigurationOperation from a given name.
// The name must be that of a previously created DeactivateConfigurationOperation, possibly from a different
process.
func (c *gRPCClient) DeactivateConfigurationOperation(name string) *DeactivateConfigurationOperation {
return &DeactivateConfigurationOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
}
}

```

```

// DeactivateConfigurationOperation returns a new DeactivateConfigurationOperation from a given name.
// The name must be that of a previously
// created DeactivateConfigurationOperation, possibly from a different process.
func (c *restClient) DeactivateConfigurationOperation(name string) *DeactivateConfigurationOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &DeactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

// DeleteConfigurationOperation returns a new DeleteConfigurationOperation from a given name.
// The name must be that of a previously created DeleteConfigurationOperation, possibly from a different process.
func (c *gRPCClient) DeleteConfigurationOperation(name string) *DeleteConfigurationOperation {
 return &DeleteConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 }
}

// DeleteConfigurationOperation returns a new DeleteConfigurationOperation from a given name.
// The name must be that of a previously created DeleteConfigurationOperation,
// possibly from a different process.
func (c *restClient) DeleteConfigurationOperation(name string) *DeleteConfigurationOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &DeleteConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

// ReactivateConfigurationOperation returns a new ReactivateConfigurationOperation from a given name.
// The name must be that of a previously created ReactivateConfigurationOperation, possibly from a different
// process.
func (c *gRPCClient) ReactivateConfigurationOperation(name string) *ReactivateConfigurationOperation {
 return &ReactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 }
}

// ReactivateConfigurationOperation returns a new ReactivateConfigurationOperation from a given name.
// The name must be that of a previously created ReactivateConfigurationOperation, possibly
// from a different process.
func (c *restClient) ReactivateConfigurationOperation(name string) *ReactivateConfigurationOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &ReactivateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

```



```

}

// UpdateConfigurationOperation returns a new UpdateConfigurationOperation from a given name.
// The name must be that of a previously created UpdateConfigurationOperation, possibly from a different process.
func (c *gRPCClient) UpdateConfigurationOperation(name string) *UpdateConfigurationOperation {
 return &UpdateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 }
}

// UpdateConfigurationOperation returns a new UpdateConfigurationOperation from a given name.
// The name must be that of a previously created UpdateConfigurationOperation, possibly from a different process.
func (c
 *restClient) UpdateConfigurationOperation(name string) *UpdateConfigurationOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &UpdateConfigurationOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

// Copyright 2024 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

```

```

//go:build go1.23

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleLicensesClient_List_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications
 to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
 }
 for resp, err := range c.List(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,

```

```

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license_service.proto

package discoveryenginepb

import (
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 _ "google.golang.org/genproto/googleapis/api/annotations"
 status "google.golang.org/genproto/googleapis/rpc/status"
 protoreflect
 "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 fieldmaskpb "google.golang.org/protobuf/types/known/fieldmaskpb"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect "reflect"
 sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Request message for
// [UserService.ListUserLicenses][google.cloud.discoveryengine.v1.UserService.ListUserLicenses].
type ListUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The parent [UserStore][] resource name, format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 // Parent string `protobuf:"bytes,1,opt,name=parent,proto3"
 // json:"parent,omitempty"`
 // Optional. Requested page size. Server may return fewer items than
 // requested. If unspecified, defaults to 10. The maximum value is 50; values
 // above 50 will be coerced to 50.
 //
 // If this field is negative, an INVALID_ARGUMENT error is returned.

```

```

PageSize int32 `protobuf:"varint,2,opt,name=page_size,json=pageSize,proto3" json:"page_size,omitempty"`
// Optional. A page token, received from a previous `ListUserLicenses` call.
// Provide this to retrieve the subsequent page.
//
// When paginating, all other parameters provided to `ListUserLicenses`
// must match the call that provided the page token.
PageToken string `protobuf:"bytes,3,opt,name=page_token,json=pageToken,proto3" json:"page_token,omitempty"`
// Optional. Filter for the list request.
//
// Supported fields:
//
// * `license_assignment_state`
//
// Examples:
//
// * `license_assignment_state = ASSIGNED` to list assigned user licenses.
// * `license_assignment_state
= NO_LICENSE` to list not licensed users.
// * `license_assignment_state = NO_LICENSE_ATTEMPTED_LOGIN` to list users
// who attempted login but no license assigned.
// * `license_assignment_state != NO_LICENSE_ATTEMPTED_LOGIN` to filter
// out users who attempted login but no license assigned.
Filter string `protobuf:"bytes,4,opt,name=filter,proto3" json:"filter,omitempty"`
}

func (x *ListUserLicensesRequest) Reset() {
*x = ListUserLicensesRequest{}
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *ListUserLicensesRequest) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*ListUserLicensesRequest) ProtoMessage() {}

func (x *ListUserLicensesRequest) ProtoReflect() protoreflect.Message {
mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if
ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
}

```

```

return mi.MessageOf(x)
}

// Deprecated: Use ListUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*ListUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{0}
}

func (x *ListUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *ListUserLicensesRequest) GetPageSize() int32 {
 if x != nil {
 return x.PageSize
 }
 return 0
}

func (x *ListUserLicensesRequest) GetPageToken() string {
 if x != nil {
 return x.PageToken
 }
 return ""
}

func (x *ListUserLicensesRequest) GetFilter() string {
 if x != nil {
 return x.Filter
 }
 return ""
}

// Response message for
// [UserLicenseService.ListUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses].
type ListUserLicensesResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // All the customer's
 // [UserLicense][google.cloud.discoveryengine.v1.UserLicense]s.
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
}

```

```
// A token, which can be sent as `page_token` to retrieve the next page. If
// this field is omitted, there are no subsequent pages.
NextPageToken string `protobuf:"bytes,2,opt,name=next_page_token,json=nextPageToken,proto3"
json:"next_page_token,omitempty"`
}
```

```
func (x *ListUserLicensesResponse) Reset() {
 *x = ListUserLicensesResponse{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}
```

```
func (x *ListUserLicensesResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}
```

```
func (*ListUserLicensesResponse) ProtoMessage() {}
```

```
func (x *ListUserLicensesResponse) ProtoReflect() protoreflect.Message
{
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[1]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}
```

// Deprecated: Use ListUserLicensesResponse.ProtoReflect.Descriptor instead.

```
func (*ListUserLicensesResponse) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{1}
}
```

```
func (x *ListUserLicensesResponse) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}
```

```
func (x *ListUserLicensesResponse) GetNextPageToken() string {
 if x != nil {
 return x.NextPageToken
 }
}
```

```

return ""
}

// Request message for
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdat
eUserLicenses]
// method.
type BatchUpdateUserLicensesRequest struct {
 state protoimpl.MessageState
 sizeCache
 protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. The source of the input.
 //
 // Types that are assignable to Source:
 //
 // *BatchUpdateUserLicensesRequest_InlineSource_
 Source isBatchUpdateUserLicensesRequest_Source `protobuf_oneof:"source"`
 // Required. The parent [UserStore][] resource name, format:
 // `projects/{project}/locations/{location}/userStores/{user_store_id}`.
 Parent string `protobuf:"bytes,1,opt,name=parent,proto3" json:"parent,omitempty"`
 // Optional. If true, if user licenses removed associated license config, the
 // user license will be deleted. By default which is false, the user license
 // will be updated to unassigned state.
 DeleteUnassignedUserLicenses bool
 `protobuf:"varint,4,opt,name=delete_unassigned_user_licenses,json=deleteUnassignedUserLicenses,proto3"
 json:"delete_unassigned_user_licenses,omitempty"`
}

func (x *BatchUpdateUserLicensesRequest) Reset() {
 *x = BatchUpdateUserLicensesRequest{}
 mi
 := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest) ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2]
 if x != nil {

```

```

ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{2}
}

func (m *BatchUpdateUserLicensesRequest) GetSource() isBatchUpdateUserLicensesRequest_Source {
 {
 if m != nil {
 return m.Source
 }
 return nil
 }
}

func (x *BatchUpdateUserLicensesRequest) GetInlineSource() *BatchUpdateUserLicensesRequest_InlineSource {
 if x, ok := x.GetSource().(*BatchUpdateUserLicensesRequest_InlineSource); ok {
 return x.InlineSource
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest) GetParent() string {
 if x != nil {
 return x.Parent
 }
 return ""
}

func (x *BatchUpdateUserLicensesRequest) GetDeleteUnassignedUserLicenses() bool {
 if x != nil {
 return x.DeleteUnassignedUserLicenses
 }
 return false
}

type isBatchUpdateUserLicensesRequest_Source interface {
 isBatchUpdateUserLicensesRequest_Source()
}

type BatchUpdateUserLicensesRequest_InlineSource_struct {

```



```

// The inline source for the input content for document embeddings.
InlineSource *BatchUpdateUserLicensesRequest_InlineSource
`protobuf:"bytes,2,opt,name=inline_source,json=inlineSource,proto3,oneof"`
}

func (*BatchUpdateUserLicensesRequest_InlineSource_) isBatchUpdateUserLicensesRequest_Source()
{}

// Metadata related to the progress of the
//
[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses]
// operation. This will be returned by the google.longrunning.Operation.metadata
// field.
type BatchUpdateUserLicensesMetadata struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Operation create time.
 CreateTime *timestamppb.Timestamp `protobuf:"bytes,1,opt,name=create_time,json=createTime,proto3"
 json:"create_time,omitempty"`
 // Operation last update time. If the operation is done, this is also the
 // finish time.
 UpdateTime *timestamppb.Timestamp `protobuf:"bytes,2,opt,name=update_time,json=updateTime,proto3"
 json:"update_time,omitempty"`
 // Count of user licenses successfully updated.
 SuccessCount int64 `protobuf:"varint,3,opt,name=success_count,json=successCount,proto3"
 json:"success_count,omitempty"`
 // Count
 // of user licenses that failed to be updated.
 FailureCount int64 `protobuf:"varint,4,opt,name=failure_count,json=failureCount,proto3"
 json:"failure_count,omitempty"`
}

func (x *BatchUpdateUserLicensesMetadata) Reset() {
 *x = BatchUpdateUserLicensesMetadata{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesMetadata) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesMetadata) ProtoMessage() {}

```

```

func (x *BatchUpdateUserLicensesMetadata) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[3]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesMetadata.ProtoReflect.Descriptor
// instead.
func (*BatchUpdateUserLicensesMetadata) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{3}
}

func (x *BatchUpdateUserLicensesMetadata) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}

func (x *BatchUpdateUserLicensesMetadata) GetSuccessCount() int64 {
 if x != nil {
 return x.SuccessCount
 }
 return 0
}

func (x *BatchUpdateUserLicensesMetadata) GetFailureCount() int64 {
 if x != nil {
 return x.FailureCount
 }
 return 0
}

// Response message for
//

```

```

[UserLicenseService.BatchUpdateUserLicenses][google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses]
// method.
type BatchUpdateUserLicensesResponse struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields
 protoimpl.UnknownFields

 // UserLicenses successfully updated.
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
 // A sample of errors encountered while processing the request.
 ErrorSamples []*status.Status `protobuf:"bytes,2,rep,name=error_samples,json=errorSamples,proto3"
 json:"error_samples,omitempty"`
}

func (x *BatchUpdateUserLicensesResponse) Reset() {
 *x = BatchUpdateUserLicensesResponse{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesResponse) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesResponse) ProtoMessage() {}

func (x *BatchUpdateUserLicensesResponse) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[4]
 if x != nil {
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesResponse.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesResponse) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{4}
}

```

```

func (x *BatchUpdateUserLicensesResponse) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *BatchUpdateUserLicensesResponse) GetErrorSamples() []*status.Status {
 if x != nil {
 return x.ErrorSamples
 }
 return nil
}

// The inline source for the input config for BatchUpdateUserLicenses
// method.
type BatchUpdateUserLicensesRequest_InlineSource struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Required. A list of user licenses to update. Each user license
 //
 // must have a valid
 // [UserLicense.user_principal][google.cloud.discoveryengine.v1.UserLicense.user_principal].
 UserLicenses []*UserLicense `protobuf:"bytes,1,rep,name=user_licenses,json=userLicenses,proto3"
 json:"user_licenses,omitempty"`
 // Optional. The list of fields to update.
 UpdateMask *fieldmaskpb.FieldMask `protobuf:"bytes,2,opt,name=update_mask,json=updateMask,proto3"
 json:"update_mask,omitempty"`
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) Reset() {
 *x = BatchUpdateUserLicensesRequest_InlineSource{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*BatchUpdateUserLicensesRequest_InlineSource) ProtoMessage() {}

func (x *BatchUpdateUserLicensesRequest_InlineSource) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[5]
 if

```

```

x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use BatchUpdateUserLicensesRequest_InlineSource.ProtoReflect.Descriptor instead.
func (*BatchUpdateUserLicensesRequest_InlineSource) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP(), []int{2, 0}
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUserLicenses() []*UserLicense {
 if x != nil {
 return x.UserLicenses
 }
 return nil
}

func (x *BatchUpdateUserLicensesRequest_InlineSource) GetUpdateMask() *fieldmaskpb.FieldMask {
 if x != nil {
 return x.UpdateMask
 }
 return nil
}

var File_google_cloud_discoveryengine_v1_user_license_service_proto protoreflect.FileDescriptor

var file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc = []byte{
 0x0a, 0x3a, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,
 0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x73,
 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f,
 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f,
 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1c, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x61, 0x6e, 0x6e, 0x6f, 0x74, 0x61,
 0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x63, 0x6c, 0x69, 0x65, 0x6e, 0x74, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
 0x2f, 0x66, 0x66, 0x69,
 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e,
 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,
 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f,
 0x1a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,

```

0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,  
0x31, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,  
0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x6c, 0x6f, 0x6e,  
0x67, 0x72, 0x75, 0x6e, 0x6e, 0x69, 0x6e, 0x67, 0x2f, 0x6f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69,  
0x6f, 0x6e, 0x73, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x20, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64,  
0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x2e, 0x70, 0x72,  
0x6f, 0x74, 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,  
0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d,  
0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x17, 0x67, 0x6f,  
0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x72, 0x70, 0x63, 0x2f, 0x73, 0x74, 0x61, 0x74, 0x75, 0x73, 0x2e,  
0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0xc6, 0x01, 0x0a, 0x17, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x12, 0x48, 0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x30, 0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f,  
0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,  
0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74,  
0x6f, 0x72, 0x65, 0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x20,  
0x0a, 0x09, 0x70,  
0x61, 0x67, 0x65, 0x5f, 0x73, 0x69, 0x7a, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x05, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x08, 0x70, 0x61, 0x67, 0x65, 0x53, 0x69, 0x7a, 0x65, 0x12, 0x22, 0x0a,  
0x0a, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65, 0x6e, 0x18, 0x03, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x09, 0x70, 0x61, 0x67, 0x65, 0x54, 0x6f, 0x6b, 0x65,  
0x6e, 0x12, 0x1b, 0x0a, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x06, 0x66, 0x69, 0x6c, 0x74, 0x65, 0x72, 0x22, 0x95,  
0x01, 0x0a, 0x18, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,  
0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75,  
0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03,  
0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75,  
0x64, 0x2e,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e,  
0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,  
0x52, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x26,  
0x0a, 0x0f, 0x6e, 0x65, 0x78, 0x74, 0x5f, 0x70, 0x61, 0x67, 0x65, 0x5f, 0x74, 0x6f, 0x6b, 0x65,  
0x6e, 0x18, 0x02, 0x20, 0x01, 0x28, 0x09, 0x52, 0x0d, 0x6e, 0x65, 0x78, 0x74, 0x50, 0x61, 0x67,  
0x65, 0x54, 0x6f, 0x6b, 0x65, 0x6e, 0x22, 0xe0, 0x03, 0x0a, 0x1e, 0x42, 0x61, 0x74, 0x63, 0x68,  
0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x12, 0x73, 0x0a, 0x0d, 0x69, 0x6e, 0x6c,  
0x69, 0x6e, 0x65, 0x5f, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b,  
0x32, 0x4c, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e,  
0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65,  
0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,  
0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73,  
0x74, 0x2e, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x48, 0x00,  
0x52, 0x0c, 0x69, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x48,  
0x0a, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x30,  
0xe0, 0x41, 0x02, 0xfa, 0x41, 0x2a, 0x0a, 0x28, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,

0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70,  
0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x55, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65,  
0x52, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x12, 0x4a, 0x0a, 0x1f, 0x64, 0x65, 0x6c, 0x65,  
0x74, 0x65, 0x5f, 0x75, 0x6e, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65,  
0x64, 0x5f, 0x75, 0x73,  
0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x04, 0x20, 0x01, 0x28,  
0x08, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x1c, 0x64, 0x65, 0x6c, 0x65, 0x74, 0x65, 0x55, 0x6e,  
0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x65, 0x64, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,  
0x6e, 0x73, 0x65, 0x73, 0x1a, 0xa8, 0x01, 0x0a, 0x0c, 0x49, 0x6e, 0x6c, 0x69, 0x6e, 0x65, 0x53,  
0x6f, 0x75, 0x72, 0x63, 0x65, 0x12, 0x56, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x52,  
0x0c, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x40, 0x0a,  
0x0b,  
0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x6d, 0x61, 0x73, 0x6b, 0x18, 0x02, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x46, 0x69, 0x65, 0x6c, 0x64, 0x4d, 0x61, 0x73, 0x6b, 0x42, 0x03,  
0xe0, 0x41, 0x01, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x4d, 0x61, 0x73, 0x6b, 0x42,  
0x08, 0x0a, 0x06, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x22, 0xe5, 0x01, 0x0a, 0x1f, 0x42, 0x61,  
0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63,  
0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d, 0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x12, 0x3b, 0x0a,  
0x0b, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x01, 0x20, 0x01,  
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,  
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a,  
0x63, 0x72, 0x65, 0x61, 0x74, 0x65,  
0x54, 0x69, 0x6d, 0x65, 0x12, 0x3b, 0x0a, 0x0b, 0x75, 0x70,  
0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0b, 0x32,  
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75,  
0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x52, 0x0a, 0x75, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x23, 0x0a, 0x0d, 0x73, 0x75, 0x63, 0x63, 0x65,  
0x73, 0x73, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x03, 0x20, 0x01, 0x28, 0x03, 0x52, 0x0c,  
0x73, 0x75, 0x63, 0x63, 0x65, 0x73, 0x73, 0x43, 0x6f, 0x75, 0x6e, 0x74, 0x12, 0x23, 0x0a, 0x0d,  
0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x5f, 0x63, 0x6f, 0x75, 0x6e, 0x74, 0x18, 0x04, 0x20,  
0x01, 0x28, 0x03, 0x52, 0x0c, 0x66, 0x61, 0x69, 0x6c, 0x75, 0x72, 0x65, 0x43, 0x6f, 0x75, 0x6e,  
0x74, 0x22, 0xad, 0x01, 0x0a, 0x1f, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74,  
0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x52, 0x65, 0x73,  
0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x51, 0x0a, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69,  
0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x18, 0x01, 0x20, 0x03, 0x28, 0x0b, 0x32, 0x2c, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,  
0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,  
0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x52, 0x0c, 0x75, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x37, 0x0a, 0x0d, 0x65, 0x72, 0x72, 0x6f,  
0x72, 0x5f, 0x73, 0x61, 0x6d, 0x70, 0x6c, 0x65, 0x73, 0x18, 0x02, 0x20, 0x03, 0x28, 0x0b, 0x32,  
0x12, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x72, 0x70, 0x63, 0x2e, 0x53, 0x74, 0x61,  
0x74, 0x75, 0x73, 0x52, 0x0c, 0x65, 0x72, 0x72, 0x6f, 0x72, 0x53, 0x61, 0x6d, 0x70, 0x6c, 0x65,

0x73, 0x32, 0x9a, 0x05, 0x0a, 0x12, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65,  
0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x12, 0xd7, 0x01, 0x0a, 0x10, 0x4c, 0x69, 0x73,  
0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x38, 0x2e,  
0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73,  
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e,  
0x4c, 0x69, 0x73, 0x74, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73,  
0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a, 0x39, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65,  
0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,  
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x4c, 0x69, 0x73, 0x74, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x73, 0x70, 0x6f, 0x6e,  
0x73, 0x65, 0x22, 0x4e, 0xda, 0x41, 0x06, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x82, 0xd3, 0xe4,  
0x93, 0x02, 0x3f, 0x12, 0x3d,  
0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74,  
0x3d, 0x70, 0x72, 0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61,  
0x74, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72,  
0x65, 0x73, 0x2f, 0x2a, 0x7d, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73,  
0x65, 0x73, 0x12, 0xd5, 0x02, 0x0a, 0x17, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x12, 0x3f,  
0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69,  
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31,  
0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72,  
0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65, 0x71, 0x75, 0x65, 0x73, 0x74, 0x1a,  
0x1d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x6c,  
0x6f, 0x6e, 0x67, 0x72, 0x75, 0x6e,  
0x6e, 0x69, 0x6e, 0x67, 0x2e, 0x4f, 0x70, 0x65, 0x72, 0x61, 0x74, 0x69, 0x6f, 0x6e, 0x22, 0xd9,  
0x01, 0xca, 0x41, 0x82, 0x01, 0x0a, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,  
0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,  
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61,  
0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x52, 0x65,  
0x73, 0x70, 0x6f, 0x6e, 0x73, 0x65, 0x12, 0x3f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63,  
0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,  
0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x42, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64,  
0x61, 0x74, 0x65, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x4d,  
0x65, 0x74, 0x61, 0x64, 0x61, 0x74, 0x61, 0x82, 0xd3, 0xe4, 0x93, 0x02, 0x4d, 0x3a, 0x01,  
0x2a,  
0x22, 0x48, 0x2f, 0x76, 0x31, 0x2f, 0x7b, 0x70, 0x61, 0x72, 0x65, 0x6e, 0x74, 0x3d, 0x70, 0x72,  
0x6f, 0x6a, 0x65, 0x63, 0x74, 0x73, 0x2f, 0x2a, 0x2f, 0x6c, 0x6f, 0x63, 0x61, 0x74, 0x69, 0x6f,  
0x6e, 0x73, 0x2f, 0x2a, 0x2f, 0x75, 0x73, 0x65, 0x72, 0x53, 0x74, 0x6f, 0x72, 0x65, 0x73, 0x2f,  
0x2a, 0x7d, 0x3a, 0x62, 0x61, 0x74, 0x63, 0x68, 0x55, 0x70, 0x64, 0x61, 0x74, 0x65, 0x55, 0x73,  
0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x73, 0x1a, 0x52, 0xca, 0x41, 0x1e, 0x64,  
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x67,  
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0xd2, 0x41, 0x2e,  
0x68, 0x74, 0x74, 0x70, 0x73, 0x3a, 0x2f, 0x2f, 0x77, 0x77, 0x77, 0x2e, 0x67, 0x6f, 0x6f, 0x67,  
0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x61, 0x75, 0x74, 0x68, 0x2f,  
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2d, 0x70, 0x6c, 0x61, 0x74, 0x66, 0x6f, 0x72, 0x6d, 0x42, 0x8a,  
0x02, 0x0a, 0x23, 0x63,



```

0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c,
0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x42, 0x17, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65,
0x6e, 0x73, 0x65, 0x53, 0x65, 0x72, 0x76, 0x69, 0x63, 0x65, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50,
0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e,
0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79,
0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64,
0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x70, 0x62,
0xa2, 0x02, 0x0f, 0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45, 0x52, 0x59, 0x45, 0x4e, 0x47, 0x49,
0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67,
0x6c, 0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75,
0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e,
0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c,
0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67,
0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a,
0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56, 0x31, 0x62, 0x06, 0x70, 0x72, 0x6f,
0x74, 0x6f, 0x33,
}

var (
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc
)

func file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescOnce.Do(func()
 {
 file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData =
 protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDescData
}

var file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes =
make([]protoimpl.MessageInfo, 6)
var file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = []any{
 (*ListUserLicensesRequest)(nil), // 0: google.cloud.discoveryengine.v1.ListUserLicensesRequest
 (*ListUserLicensesResponse)(nil), // 1: google.cloud.discoveryengine.v1.ListUserLicensesResponse
 (*BatchUpdateUserLicensesRequest)(nil), // 2:
 google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
 (*BatchUpdateUserLicensesMetadata)(nil), // 3:
 google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata
 (*BatchUpdateUserLicensesResponse)(nil), // 4: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse
 (*BatchUpdateUserLicensesRequest_InlineSource)(nil), // 5:

```

```

google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
(*UserLicense)(nil), // 6: google.cloud.discoveryengine.v1.UserLicense
(*timestamppb.Timestamp)(nil), // 7: google.protobuf.Timestamp
(*status.Status)(nil), // 8: google.rpc.Status
(*fieldmaskpb.FieldMask)(nil), // 9: google.protobuf.FieldMask
(*longrunningpb.Operation)(nil), // 10: google.longrunning.Operation
}

var file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = []int32{
 6, // 0: google.cloud.discoveryengine.v1.ListUserLicensesResponse.user_licenses:type_name ->
google.cloud.discoveryengine.v1.UserLicense
 5, // 1: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.inline_source:type_name
-> google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource
 7, // 2: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.create_time:type_name ->
google.protobuf.Timestamp
 7, // 3: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesMetadata.update_time:type_name ->
google.protobuf.Timestamp
 6, // 4: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.user_licenses:type_name ->
google.cloud.discoveryengine.v1.UserLicense
 8, // 5: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesResponse.error_samples:type_name ->
google.rpc.Status
 6, // 6: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.user_licenses:type_name -
> google.cloud.discoveryengine.v1.UserLicense
 9, // 7: google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest.InlineSource.update_mask:type_name -
> google.protobuf.FieldMask
 0, // 8: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:input_type
-> google.cloud.discoveryengine.v1.ListUserLicensesRequest
 2, // 9: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:input_type ->
google.cloud.discoveryengine.v1.BatchUpdateUserLicensesRequest
 1, // 10: google.cloud.discoveryengine.v1.UserLicenseService.ListUserLicenses:output_type ->
google.cloud.discoveryengine.v1.ListUserLicensesResponse
 10, // 11: google.cloud.discoveryengine.v1.UserLicenseService.BatchUpdateUserLicenses:output_type ->
google.longrunning.Operation
 10, // [10:12] is the sub-list for method output_type
 8, // [8:10] is the sub-list for method input_type
 8, // [8:8] is the sub-list for extension type_name
 8, // [8:8] is the sub-list for extension extendee
 0, // [0:8] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_service_proto_init() }
func file_google_cloud_discoveryengine_v1_user_license_service_proto_init() {
 if File_google_cloud_discoveryengine_v1_user_license_service_proto
 != nil {
 return
 }
 file_google_cloud_discoveryengine_v1_user_license_proto_init()
 file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes[2].OneofWrappers = []any{
 (*BatchUpdateUserLicensesRequest_InlineSource_)(nil),

```

```

}
type x struct{}
out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc,
 NumEnums: 0,
 NumMessages: 6,
 NumExtensions: 0,
 NumServices: 1,
 },
 GoTypes: file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes,
 DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs,
 MessageInfos: file_google_cloud_discoveryengine_v1_user_license_service_proto_msgTypes,
}.Build()
File_google_cloud_discoveryengine_v1_user_license_service_proto = out.File
file_google_cloud_discoveryengine_v1_user_license_service_proto_rawDesc
= nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_goTypes = nil
file_google_cloud_discoveryengine_v1_user_license_service_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// Code generated by protoc-gen-go_gapic. DO NOT EDIT.
//go:build go1.23
package licensemanager_test
import (
 "context"
 licensemanager "cloud.google.com/go/licensemanager/apiv1"
 licensemanagerpb "cloud.google.com/go/licensemanager/apiv1/licensemanagerpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 locationpb "google.golang.org/genproto/googleapis/cloud/location"

```

)

```
func ExampleClient_AggregateUsage_all()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.AggregateUsageRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#AggregateUsageRequest.
 }
 for resp, err := range c.AggregateUsage(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

func ExampleClient_ListConfigurations_all() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.ListConfigurationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ListConfigurationsRequest.
 }
 for resp, err := range c.ListConfigurations(ctx, req).All() {
```

```

if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
}
// TODO: Use resp.
_ = resp
}
}

func ExampleClient_ListInstances_all() {
 ctx := context.Background()
 // This
 snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &licensemanagerpb.ListInstancesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licensemanager/apiv1/licensemanagerpb#ListInstancesRequest.
 }
 for resp, err := range c.ListInstances(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

func ExampleClient_ListProducts_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and
 should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licensemanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()
}

```

```

req := &licenseanagerpb.ListProductsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/licenseanager/apiv1/licenseanagerpb#ListProductsRequest.
}
for resp, err := range c.ListProducts(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}

```

```

func ExampleClient_ListLocations_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It
 will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := licenseanager.NewClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &locationpb.ListLocationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/google.golang.org/genproto/googleapis/cloud/location#ListLocationsRequest.
}
for resp, err := range c.ListLocations(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}

```

```

func ExampleClient_ListOperations_all() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require
 correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:

```

```

// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := licensemanager.NewClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.ListOperationsRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
}
for resp, err := range c.ListOperations(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
}
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/cloud/discoveryengine/v1/user_license.proto

package discoveryenginepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 timestamppb "google.golang.org/protobuf/types/known/timestamppb"
 reflect

```

```

"reflect"
sync "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// License assignment state enumeration.
type UserLicense_LicenseAssignmentState int32

const (
 // Default value.
 UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED UserLicense_LicenseAssignmentState = 0
 // License assigned to the user.
 UserLicense_ASSIGNED UserLicense_LicenseAssignmentState = 1
 // No license assigned to the user.
 // Deprecated, translated to NO_LICENSE.
 UserLicense_UNASSIGNED UserLicense_LicenseAssignmentState = 2
 // No license assigned to the user.
 UserLicense_NO_LICENSE UserLicense_LicenseAssignmentState = 3
 // User attempted to login but no license assigned to the user.
 // This state is only used for no user first time login attempt
 // but cannot
 // get license assigned.
 // Users already logged in but cannot get license assigned will be assigned
 // NO_LICENSE state(License could be unassigned by admin).
 UserLicense_NO_LICENSE_ATTEMPTED_LOGIN UserLicense_LicenseAssignmentState = 4
)

// Enum value maps for UserLicense_LicenseAssignmentState.
var (
 UserLicense_LicenseAssignmentState_name = map[int32]string{
 0: "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED",
 1: "ASSIGNED",
 2: "UNASSIGNED",
 3: "NO_LICENSE",
 4: "NO_LICENSE_ATTEMPTED_LOGIN",
 }
 UserLicense_LicenseAssignmentState_value = map[string]int32{
 "LICENSE_ASSIGNMENT_STATE_UNSPECIFIED": 0,
 "ASSIGNED": 1,
 "UNASSIGNED": 2,
 "NO_LICENSE": 3,
 "NO_LICENSE_ATTEMPTED_LOGIN": 4,
 }
)

```



```

}
)

func (x UserLicense_LicenseAssignmentState) Enum() *UserLicense_LicenseAssignmentState {
 p := new(UserLicense_LicenseAssignmentState)
 *p = x
 return p
}

func (x UserLicense_LicenseAssignmentState)
String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (UserLicense_LicenseAssignmentState) Descriptor() protoreflect.EnumDescriptor {
 return file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0].Descriptor()
}

func (UserLicense_LicenseAssignmentState) Type() protoreflect.EnumType {
 return &file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes[0]
}

func (x UserLicense_LicenseAssignmentState) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use UserLicense_LicenseAssignmentState.Descriptor instead.
func (UserLicense_LicenseAssignmentState) EnumDescriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0, 0}
}

// User License information assigned by the admin.
type UserLicense struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 //
 // Required. Immutable. The user principal of the User, could be email address
 // or other principal identifier. This field is immutable. Admin assign
 // licenses based on the user principal.
 UserPrincipal string `protobuf:"bytes,1,opt,name=user_principal,json=userPrincipal,proto3"
 json:"user_principal,omitempty"`
 // Optional. The user profile.
 // We use user full name(First name + Last name) as user profile.
 UserProfile string `protobuf:"bytes,3,opt,name=user_profile,json=userProfile,proto3"
 json:"user_profile,omitempty"`
 // Output only. License assignment state of the user.

```

```

// If the user is assigned with a license config, the user login will be
// assigned with the license;
// If the user's license assignment state is unassigned or unspecified, no
// license config will be associated to the user;
LicenseAssignmentState UserLicense_LicenseAssignmentState
`protobuf:"varint,4,opt,name=license_assignment_state,json=licenseAssignmentState,proto3,enum=google.cloud.discoveryengine.v1.UserLicense_LicenseAssignmentState"
json:"license_assignment_state,omitempty"`
// Optional. The full resource name of the Subscription(LicenseConfig)
// assigned to the user.
LicenseConfig string `protobuf:"bytes,5,opt,name=license_config,json=licenseConfig,proto3"
json:"license_config,omitempty"`
// Output only. User created timestamp.
CreateTime *timestamppb.Timestamp `protobuf:"bytes,6,opt,name=create_time,json=createTime,proto3"
json:"create_time,omitempty"`
// Output only. User update timestamp.
UpdateTime *timestamppb.Timestamp `protobuf:"bytes,7,opt,name=update_time,json=updateTime,proto3"
json:"update_time,omitempty"`
// Output only. User last logged in time.
// If the user has not logged in yet, this field will be empty.
LastLoginTime *timestamppb.Timestamp
`protobuf:"bytes,8,opt,name=last_login_time,json=lastLoginTime,proto3" json:"last_login_time,omitempty"`
}

func (x *UserLicense) Reset() {
 *x = UserLicense{}
 mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
 ms
 := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 ms.StoreMessageInfo(mi)
}

func (x *UserLicense) String() string {
 return protoimpl.X.MessageStringOf(x)
}

func (*UserLicense) ProtoMessage() {}

func (x *UserLicense) ProtoReflect() protoreflect.Message {
 mi := &file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes[0]
 if x != nil {
 ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
 if ms.LoadMessageInfo() == nil {
 ms.StoreMessageInfo(mi)
 }
 return ms
 }
 return mi.MessageOf(x)
}

```

```

}

// Deprecated: Use UserLicense.ProtoReflect.Descriptor instead.
func (*UserLicense) Descriptor() ([]byte, []int) {
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP(), []int{0}
}

func (x *UserLicense) GetUserPrincipal() string {
 if x != nil {
 return x.UserPrincipal
 }
 return ""
}

func (x *UserLicense) GetUserProfile() string {
 if x != nil {
 return x.UserProfile
 }
 return ""
}

func (x *UserLicense) GetLicenseAssignmentState()
 UserLicense_LicenseAssignmentState {
 if x != nil {
 return x.LicenseAssignmentState
 }
 return UserLicense_LICENSE_ASSIGNMENT_STATE_UNSPECIFIED
}

func (x *UserLicense) GetLicenseConfig() string {
 if x != nil {
 return x.LicenseConfig
 }
 return ""
}

func (x *UserLicense) GetCreateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.CreateTime
 }
 return nil
}

func (x *UserLicense) GetUpdateTime() *timestamppb.Timestamp {
 if x != nil {
 return x.UpdateTime
 }
 return nil
}

```

```
}
```

```
func (x *UserLicense) GetLastLoginTime() *timestamppb.Timestamp {
 if x != nil {
 return x.LastLoginTime
 }
 return nil
}
```

```
var File_google_cloud_discoveryengine_v1_user_license_proto protoreflect.FileDescriptor
```

```
var file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = []byte{
 0x0a, 0x32, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2f, 0x64,
 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x76,
 0x31,
 0x2f, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e, 0x70,
 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f,
 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,
 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70,
 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65, 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61,
 0x70, 0x69, 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,
 0x6f, 0x1a, 0x1f, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,
 0x75, 0x66, 0x2f, 0x74, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x2e, 0x70, 0x72, 0x6f,
 0x74, 0x6f, 0x22, 0xa6, 0x05, 0x0a, 0x0b, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e,
 0x73, 0x65, 0x12, 0x2d, 0x0a, 0x0e,
 0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x69, 0x6e, 0x63,
 0x69, 0x70, 0x61, 0x6c, 0x18, 0x01, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x05, 0xe0,
 0x41, 0x02, 0x52, 0x0d, 0x75, 0x73, 0x65, 0x72, 0x50, 0x72, 0x69, 0x6e, 0x63, 0x69, 0x70, 0x61,
 0x6c, 0x12, 0x26, 0x0a, 0x0c, 0x75, 0x73, 0x65, 0x72, 0x5f, 0x70, 0x72, 0x6f, 0x66, 0x69, 0x6c,
 0x65, 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x03, 0xe0, 0x41, 0x01, 0x52, 0x0b, 0x75, 0x73,
 0x65, 0x72, 0x50, 0x72, 0x6f, 0x66, 0x69, 0x6c, 0x65, 0x12, 0x82, 0x01, 0x0a, 0x18, 0x6c, 0x69,
 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x61, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74,
 0x5f, 0x73, 0x74, 0x61, 0x74, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x43, 0x2e, 0x67,
 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x64, 0x69, 0x73, 0x63,
 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x76, 0x31, 0x2e, 0x55,
 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x2e,
 0x4c, 0x69, 0x63, 0x65, 0x6e,
 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74,
 0x65, 0x42, 0x03, 0xe0, 0x41, 0x03, 0x52, 0x16, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x41,
 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74, 0x61, 0x74, 0x65, 0x12, 0x5b,
 0x0a, 0x0e, 0x6c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x5f, 0x63, 0x6f, 0x6e, 0x66, 0x69, 0x67,
 0x18, 0x05, 0x20, 0x01, 0x28, 0x09, 0x42, 0x34, 0xe0, 0x41, 0x01, 0xfa, 0x41, 0x2e, 0x0a, 0x2c,
 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e,
 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x4c,
 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x52, 0x0d, 0x6c, 0x69,
 0x63, 0x65, 0x6e, 0x73, 0x65, 0x43, 0x6f, 0x6e, 0x66, 0x69, 0x67, 0x12, 0x40, 0x0a, 0x0b, 0x63,
 0x72, 0x65, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x06, 0x20, 0x01, 0x28, 0x0b,
```

```

0x32,
0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62,
0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03, 0xe0, 0x41,
0x03, 0x52, 0x0a, 0x63, 0x72, 0x65, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12, 0x40, 0x0a,
0x0b, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x5f, 0x74, 0x69, 0x6d, 0x65, 0x18, 0x07, 0x20, 0x01,
0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74,
0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73, 0x74, 0x61, 0x6d, 0x70, 0x42, 0x03,
0xe0, 0x41, 0x03, 0x52, 0x0a, 0x75, 0x70, 0x64, 0x61, 0x74, 0x65, 0x54, 0x69, 0x6d, 0x65, 0x12,
0x47, 0x0a, 0x0f, 0x6c, 0x61, 0x73, 0x74, 0x5f, 0x6c, 0x6f, 0x67, 0x69, 0x6e, 0x5f, 0x74, 0x69,
0x6d, 0x65, 0x18, 0x08, 0x20, 0x01, 0x28, 0x0b, 0x32, 0x1a, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x62, 0x75, 0x66, 0x2e, 0x54, 0x69, 0x6d, 0x65, 0x73,
0x74, 0x61, 0x6d, 0x70, 0x42,
0x03, 0xe0, 0x41, 0x03, 0x52, 0x0d, 0x6c, 0x61, 0x73, 0x74, 0x4c,
0x6f, 0x67, 0x69, 0x6e, 0x54, 0x69, 0x6d, 0x65, 0x22, 0x90, 0x01, 0x0a, 0x16, 0x4c, 0x69, 0x63,
0x65, 0x6e, 0x73, 0x65, 0x41, 0x73, 0x73, 0x69, 0x67, 0x6e, 0x6d, 0x65, 0x6e, 0x74, 0x53, 0x74,
0x61, 0x74, 0x65, 0x12, 0x28, 0x0a, 0x24, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41,
0x53, 0x53, 0x49, 0x47, 0x4e, 0x4d, 0x45, 0x4e, 0x54, 0x5f, 0x53, 0x54, 0x41, 0x54, 0x45, 0x5f,
0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44, 0x10, 0x00, 0x12, 0x0c, 0x0a,
0x08, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x01, 0x12, 0x0e, 0x0a, 0x0a, 0x55,
0x4e, 0x41, 0x53, 0x53, 0x49, 0x47, 0x4e, 0x45, 0x44, 0x10, 0x02, 0x12, 0x0e, 0x0a, 0x0a, 0x4e,
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x10, 0x03, 0x12, 0x1e, 0x0a, 0x1a, 0x4e,
0x4f, 0x5f, 0x4c, 0x49, 0x43, 0x45, 0x4e, 0x53, 0x45, 0x5f, 0x41, 0x54, 0x54, 0x45, 0x4d, 0x50,
0x54, 0x45, 0x44, 0x5f, 0x4c, 0x4f, 0x47, 0x49, 0x4e, 0x10,
0x04, 0x42, 0x83, 0x02, 0x0a, 0x23,
0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6c, 0x6f, 0x75, 0x64,
0x2e, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65,
0x2e, 0x76, 0x31, 0x42, 0x10, 0x55, 0x73, 0x65, 0x72, 0x4c, 0x69, 0x63, 0x65, 0x6e, 0x73, 0x65,
0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x4d, 0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d, 0x2f, 0x67, 0x6f, 0x2f, 0x64, 0x69, 0x73,
0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2f, 0x61, 0x70, 0x69,
0x76, 0x31, 0x2f, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e, 0x67, 0x69,
0x6e, 0x65, 0x70, 0x62, 0x3b, 0x64, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x65, 0x6e,
0x67, 0x69, 0x6e, 0x65, 0x70, 0x62, 0xa2, 0x02, 0x0f, 0x44, 0x49, 0x53, 0x43, 0x4f, 0x56, 0x45,
0x52, 0x59, 0x45, 0x4e, 0x47, 0x49, 0x4e, 0x45, 0xaa, 0x02, 0x1f, 0x47, 0x6f, 0x6f, 0x67,
0x6c,
0x65, 0x2e, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76, 0x65, 0x72,
0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x2e, 0x56, 0x31, 0xca, 0x02, 0x1f, 0x47, 0x6f, 0x6f,
0x67, 0x6c, 0x65, 0x5c, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x5c, 0x44, 0x69, 0x73, 0x63, 0x6f, 0x76,
0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x5c, 0x56, 0x31, 0xea, 0x02, 0x22, 0x47,
0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x3a, 0x3a, 0x43, 0x6c, 0x6f, 0x75, 0x64, 0x3a, 0x3a, 0x44, 0x69,
0x73, 0x63, 0x6f, 0x76, 0x65, 0x72, 0x79, 0x45, 0x6e, 0x67, 0x69, 0x6e, 0x65, 0x3a, 0x3a, 0x56,
0x31, 0x62, 0x06, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce sync.Once
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc

```

)

```
func file_google_cloud_discoveryengine_v1_user_license_proto_rawDescGZIP() []byte {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescOnce.Do(func()
 {
 file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData)
 })
 return file_google_cloud_discoveryengine_v1_user_license_proto_rawDescData
}

var file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes = make([]protoimpl.EnumInfo, 1)
var file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = []any{
 (UserLicense_LicenseAssignmentState)(0), // 0:
google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 (*UserLicense)(nil), // 1: google.cloud.discoveryengine.v1.UserLicense
 (*timestamppb.Timestamp)(nil), // 2: google.protobuf.Timestamp
}
var file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = []int32{
 0, // 0: google.cloud.discoveryengine.v1.UserLicense.license_assignment_state:type_name
-> google.cloud.discoveryengine.v1.UserLicense.LicenseAssignmentState
 2, // 1: google.cloud.discoveryengine.v1.UserLicense.create_time:type_name -> google.protobuf.Timestamp
 2, // 2: google.cloud.discoveryengine.v1.UserLicense.update_time:type_name -> google.protobuf.Timestamp
 2, // 3: google.cloud.discoveryengine.v1.UserLicense.last_login_time:type_name -> google.protobuf.Timestamp
 4, // [4:4] is the sub-list for method output_type
 4, // [4:4] is the sub-list for method input_type
 4, // [4:4] is the sub-list for extension type_name
 4, // [4:4] is the sub-list for extension extendee
 0, // [0:4] is the sub-list for field type_name
}

func init() { file_google_cloud_discoveryengine_v1_user_license_proto_init() }
func file_google_cloud_discoveryengine_v1_user_license_proto_init() {
 if File_google_cloud_discoveryengine_v1_user_license_proto != nil {
 return
 }
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor:
file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc,
 NumEnums: 1,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 }
```

```

GoTypes: file_google_cloud_discoveryengine_v1_user_license_proto_goTypes,
DependencyIndexes: file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs,
EnumInfos: file_google_cloud_discoveryengine_v1_user_license_proto_enumTypes,
MessageInfos: file_google_cloud_discoveryengine_v1_user_license_proto_msgTypes,
}.Build()
File_google_cloud_discoveryengine_v1_user_license_proto = out.File
file_google_cloud_discoveryengine_v1_user_license_proto_rawDesc = nil
file_google_cloud_discoveryengine_v1_user_license_proto_goTypes = nil
file_google_cloud_discoveryengine_v1_user_license_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicenseManagementClient() {
 ctx
 := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }

```

```

}
defer c.Close()

// TODO: Use client.
_ = c
}

func ExampleNewLicenseManagementRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c,
 err := procurement.NewLicenseManagementRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseManagementClient_Assign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.AssignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#AssignRequest.
 }
 resp, err := c.Assign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 //

```



TODO: Use resp.

```
_ = resp
}
```

```
func ExampleLicenseManagementClient_EnumerateLicensedUsers() {
```

```
 ctx := context.Background()
```

```
 // This snippet has been automatically generated and should be regarded as a code template only.
```

```
 // It will require modifications to work:
```

```
 // - It may require correct/in-range values for request initialization.
```

```
 // - It may require specifying regional endpoints when creating the service client as shown in:
```

```
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
```

```
 c, err := procurement.NewLicenseManagementClient(ctx)
```

```
 if err != nil {
```

```
 // TODO: Handle error.
```

```
 }
```

```
 defer c.Close()
```

```
 req := &procurementpb.EnumerateLicensedUsersRequest{
```

```
 // TODO: Fill request struct fields.
```

```
 // See
```

```
https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicensedUsersRequest.
```

```
 }
```

```
 it := c.EnumerateLicensedUsers(ctx, req)
```

```
 for {
```

```
 resp, err := it.Next()
```

```
 if err == iterator.Done {
```

```
 break
```

```
 }
```

```
 if err != nil {
```

```
 // TODO: Handle error.
```

```
 }
```

```
 //
```

```
 TODO: Use resp.
```

```
 _ = resp
```

```
 // If you need to access the underlying RPC response,
```

```
 // you can do so by casting the `Response` as below.
```

```
 // Otherwise, remove this line. Only populated after
```

```
 // first call to Next(). Not safe for concurrent access.
```

```
 _ = it.Response.(*procurementpb.EnumerateLicensedUsersResponse)
```

```
 }
```

```
}
```

```
func ExampleLicenseManagementClient_GetLicensePool() {
```

```
 ctx := context.Background()
```

```
 // This snippet has been automatically generated and should be regarded as a code template only.
```

```
 // It will require modifications to work:
```

```

// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := procurement.NewLicenseManagementClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &procurementpb.GetLicensePoolRequest{
 // TODO: Fill request struct fields.
 //
 See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#GetLicensePoolRe
 quest.
}
resp, err := c.GetLicensePool(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseManagementClient_Unassign() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UnassignRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UnassignRequest.
 }
 resp,
 err := c.Unassign(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.

```

```

_ = resp
}

func ExampleLicenseManagementClient_UpdateLicensePool() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.UpdateLicensePoolRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#UpdateLicensePoolRequest.
 }
 resp, err := c.UpdateLicensePool(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 //
 TODO: Use resp.
 _ = resp
}

func ExampleLicenseManagementClient_GetOperation() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &longrunningpb.GetOperationRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#GetOperationRequest.
 }

```

```

resp, err := c.GetOperation(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

```

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

```
// Code generated by protoc-gen-go_gopic. DO NOT EDIT.

//go:build go1.23

package compute_test

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gopic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1beta"
 computepb "cloud.google.com/go/compute/apiv1beta/compute_pb"
 "google.golang.org/api/iterator"
)

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 //
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}
```

```

}

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetLicenseRequest.
 }

```

```

resp,
err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#GetIamPolicyLicenseRequest.
 }
 resp, err := c.GetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

```

```

req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#InsertLicenseRequest.
}
op, err := c.Insert(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded
 // as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#ListLicensesRequest.
 }
 it := c.List(ctx, req)
 for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after

```



```

// first call to Next(). Not safe for concurrent access.
-
= it.Response.(*compute.LicensesListResponse)
}
}

func ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#SetIamPolicyLicenseRequest.
 }
 resp, err := c.SetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 //
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#TestIamPermissionsLicenseRequest.
 }

```

```

}
resp, err := c.TestIamPermissions(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require
 modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1beta/compute#UpdateLicenseRequest.
 }
 op, err := c.Update(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

```

// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

 compute "cloud.google.com/go/compute/apiv1"
 computepb "cloud.google.com/go/compute/apiv1/computepb"
)

func ExampleNewLicenseCodesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

func ExampleLicenseCodesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 //
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseCodeRequest{
 // TODO: Fill request

```

```

struct fields.
// See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#GetLicenseCodeRequest.
}
resp, err := c.Get(ctx, req)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicenseCodesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicenseCodesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.TestIamPermissionsLicenseCodeRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#TestIamPermissionsLicenseCodeRequest.
 }
 resp,
 err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and

```

```

// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package procurement_test

import (
 "context"

 procurement "cloud.google.com/go/commerce/consumer/procurement/apiv1"
 procurementpb "cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb"
)

func ExampleLicenseManagementClient_EnumerateLicensedUsers_all() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := procurement.NewLicenseManagementClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &procurementpb.EnumerateLicensedUsersRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/commerce/consumer/procurement/apiv1/procurementpb#EnumerateLicensedUsersRequest.
 }
 for resp, err := range c.EnumerateLicensedUsers(ctx, req).All() {
 if err != nil {
 // TODO: Handle error and break/return/continue. Iteration will stop after any error.
 }
 // TODO: Use resp.
 _ = resp
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at

```

```

//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package discoveryengine

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 "cloud.google.com/go/longrunning"
 lroauto "cloud.google.com/go/longrunning/autogen"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
 gax
 "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 gtransport "google.golang.org/api/transport/grpc"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/grpc/codes"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var newUserLicenseClientHook clientHook

// UserLicenseCallOptions contains the retry settings for each method of UserLicenseClient.
type UserLicenseCallOptions struct {
 ListUserLicenses []gax.CallOption
 BatchUpdateUserLicenses []gax.CallOption
 CancelOperation []gax.CallOption
 GetOperation []gax.CallOption

```

```

ListOperations []gax.CallOption
}

func defaultUserLicenseGRPCClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("discoveryengine.googleapis.com:443"),
 internaloption.WithDefaultEndpointTemplate("discoveryengine.UNIVERSE_DOMAIN:443"),
 internaloption.WithDefaultMTLSEndpoint("discoveryengine.mtls.googleapis.com:443"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableJwtWithScope(),
 internaloption.EnableNewAuthLibrary(),
 option.WithGRPCDialOption(grpc.WithDefaultCallOptions(
 grpc.MaxCallRecvMsgSize(math.MaxInt32))),
}
}

func
defaultUserLicenseCallOptions() *UserLicenseCallOptions {
return &UserLicenseCallOptions{
 ListUserLicenses: []gax.CallOption{},
 BatchUpdateUserLicenses: []gax.CallOption{},
 CancelOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier:
1.30,
 })
 }),
 },
 GetOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
 },
}

```

```

ListOperations: []gax.CallOption{
 gax.WithTimeout(300000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnCodes([]codes.Code{
 codes.Unavailable,
 }, gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 30000 * time.Millisecond,
 Multiplier: 1.30,
 })
 }),
},
}

func defaultUserLicenseRESTCallOptions() *UserLicenseCallOptions {
 return &UserLicenseCallOptions{
 ListUserLicenses: []gax.CallOption{},
 BatchUpdateUserLicenses: []gax.CallOption{},
 CancelOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer
 {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusServiceUnavailable)
 }),
 },
 GetOperation: []gax.CallOption{
 gax.WithTimeout(30000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 10000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusServiceUnavailable)
 }),
 },
 ListOperations: []gax.CallOption{
 gax.WithTimeout(300000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 1000 * time.Millisecond,
 Max: 30000 * time.Millisecond,

```



```

Multiplier: 1.30,
},
http.StatusServiceUnavailable)
}),
},
}
}

// internalUserLicenseClient is an interface that defines the methods available from Discovery Engine API.
type internalUserLicenseClient
interface {
Close() error
setGoogleClientInfo(...string)
Connection() *grpc.ClientConn
ListUserLicenses(context.Context, *discoveryenginepb.ListUserLicensesRequest, ...gax.CallOption)
*UserLicenseIterator
BatchUpdateUserLicenses(context.Context, *discoveryenginepb.BatchUpdateUserLicensesRequest,
...gax.CallOption) (*BatchUpdateUserLicensesOperation, error)
BatchUpdateUserLicensesOperation(name string) *BatchUpdateUserLicensesOperation
CancelOperation(context.Context, *longrunningpb.CancelOperationRequest, ...gax.CallOption) error
GetOperation(context.Context, *longrunningpb.GetOperationRequest, ...gax.CallOption)
(*longrunningpb.Operation, error)
ListOperations(context.Context, *longrunningpb.ListOperationsRequest, ...gax.CallOption) *OperationIterator
}

// UserLicenseClient is a client for interacting with Discovery Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
//
// Service for managing
User Licenses.
type UserLicenseClient struct {
// The internal transport-dependent client.
internalClient internalUserLicenseClient

// The call options for this service.
CallOptions *UserLicenseCallOptions

// LROClient is used internally to handle long-running operations.
// It is exposed so that its CallOptions can be modified if required.
// Users should not Close this client.
LROClient *lroauto.OperationsClient
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.

```

```

func (c *UserLicenseClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *UserLicenseClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns
// a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *UserLicenseClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// ListUserLicenses lists the User Licenses.
func (c *UserLicenseClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
 return c.internalClient.ListUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicenses updates the User License.
// This method is used for batch assign/unassign licenses to users.
func (c *UserLicenseClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 return c.internalClient.BatchUpdateUserLicenses(ctx, req, opts...)
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
//
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *UserLicenseClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 return c.internalClient.BatchUpdateUserLicensesOperation(name)
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) CancelOperation(ctx context.Context, req *longrunningpb.CancelOperationRequest,
opts ...gax.CallOption) error {
 return c.internalClient.CancelOperation(ctx, req, opts...)
}

```

```

// GetOperation is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest, opts
...gax.CallOption) (*longrunningpb.Operation, error) {
 return c.internalClient.GetOperation(ctx, req, opts...)
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *UserLicenseClient) ListOperations(ctx context.Context,
 req *longrunningpb.ListOperationsRequest, opts ...gax.CallOption) *OperationIterator {
 return c.internalClient.ListOperations(ctx, req, opts...)
}

// userLicenseGRPCClient is a client for interacting with Discovery Engine API over gRPC transport.
//
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseGRPCClient struct {
 // Connection pool of gRPC connections to the service.
 connPool gtransport.ConnPool

 // Points back to the CallOptions field of the containing UserLicenseClient
 CallOptions **UserLicenseCallOptions

 // The gRPC API client.
 userLicenseClient discoveryenginepb.UserLicenseServiceClient

 // LROClient is used internally to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 operationsClient longrunningpb.OperationsClient

 // The x-goog-* metadata
 // to be sent with each request.
 xGoogHeaders []string

 logger *slog.Logger
}

// NewUserLicenseClient creates a new user license service client based on gRPC.
// The returned client must be Closed when it is done being used to clean up its underlying connections.
//
// Service for managing User Licenses.
func NewUserLicenseClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := defaultUserLicenseGRPCClientOptions()
 if newUserLicenseClientHook != nil {

```

```

hookOpts, err := newUserLicenseClientHook(ctx, clientHookParams{})
if err != nil {
 return nil, err
}
clientOpts = append(clientOpts, hookOpts...)
}

connPool, err := gtransport.DialPool(ctx, append(clientOpts, opts...)...)
if err != nil {
 return nil, err
}
client := UserLicenseClient{CallOptions: defaultUserLicenseCallOptions()}

c := &userLicenseGRPCClient{
 connPool: connPool,
 userLicenseClient: discoveryenginepb.NewUserLicenseServiceClient(connPool),
 CallOptions:
 &client.CallOptions,
 logger: internaloption.GetLogger(opts),
 operationsClient: longrunningpb.NewOperationsClient(connPool),
}
c.setGoogleClientInfo()

client.internalClient = c

client.LROClient, err = lroauto.NewOperationsClient(ctx, gtransport.WithConnPool(connPool))
if err != nil {
 // This error "should not happen", since we are just reusing old connection pool
 // and never actually need to dial.
 // If this does happen, we could leak connp. However, we cannot close conn:
 // If the user invoked the constructor with option.WithGRPCConn,
 // we would close a connection that's still in use.
 // TODO: investigate error conditions.
 return nil, err
}
c.LROClient = &client.LROClient
return &client, nil
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *userLicenseGRPCClient) Connection() *grpc.ClientConn {
 return c.connPool.Conn()
}

//

```

```

setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseGRPCClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "grpc", grpc.Version, "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *userLicenseGRPCClient) Close() error {
 return c.connPool.Close()
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
type userLicenseRESTClient struct {
 // The http endpoint to connect to.
 endpoint string

 // The http client.
 httpClient *http.Client

 // LROClient is used internally
 // to handle long-running operations.
 // It is exposed so that its CallOptions can be modified if required.
 // Users should not Close this client.
 LROClient **lroauto.OperationsClient

 // The x-goog-* headers to be sent with each request.
 xGoogHeaders []string

 // Points back to the CallOptions field of the containing UserLicenseClient
 CallOptions **UserLicenseCallOptions

 logger *slog.Logger
}

// NewUserLicenseRESTClient creates a new user license service rest client.
//
// Service for managing User Licenses.
func NewUserLicenseRESTClient(ctx context.Context, opts ...option.ClientOption) (*UserLicenseClient, error) {
 clientOpts := append(defaultUserLicenseRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {

```

```

 return nil, err
}

callOpts := defaultUserLicenseRESTCallOptions()
c := &userLicenseRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger: internaloption.GetLogger(opts),
}
c.setGoogleClientInfo()

lroOpts
:= []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
}
opClient, err := lroauto.NewOperationsRESTClient(ctx, lroOpts...)
if err != nil {
 return nil, err
}
c.LROClient = &opClient

return &UserLicenseClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultUserLicenseRESTClientOptions() []option.ClientOption {
 return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://discoveryengine.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://discoveryengine.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://discoveryengine.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://discoveryengine.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()),
 internaloption.EnableNewAuthLibrary(),
 }
}

// setGoogleClientInfo
// sets the name and version of the application in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *userLicenseRESTClient) setGoogleClientInfo(keyval ...string) {
 kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
 kv = append(kv, "gax", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
 c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
 }
}

```

```

}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *userLicenseRESTClient) Close() error {
 // Replace httpClient with nil to force cleanup.
 c.httpClient = nil
 return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *userLicenseRESTClient) Connection() *grpc.ClientConn {
 return nil
}

func (c *userLicenseGRPCClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest,
opts ...gax.CallOption) *UserLicenseIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

 hds = append(c.xGoogHeaders, hds...)
 ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
 opts =
append((*c.CallOptions).ListUserLicenses[0:len((*c.CallOptions).ListUserLicenses):len((*c.CallOptions).ListUser
Licenses)], opts...)
 it := &UserLicenseIterator{}
 req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
 it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
 resp := &discoveryenginepb.ListUserLicensesResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx,
c.userLicenseClient.ListUserLicenses, req, settings.GRPC, c.logger, "ListUserLicenses")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }
 }
}

```

```

it.Response = resp
return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
}
fetch := func(pageSize int, pageToken string) (string, error) {
items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
if err != nil {
return "", err
}
it.items = append(it.items, items...)
return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

func (c *userLicenseGRPCClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders,
hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).BatchUpdateUserLicenses[0:len((*c.CallOptions).BatchUpdateUserLicenses):len((*c.Call
Options).BatchUpdateUserLicenses)], opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
var err error
resp, err = executeRPC(ctx, c.userLicenseClient.BatchUpdateUserLicenses, req, settings.GRPC, c.logger,
"BatchUpdateUserLicenses")
return err
}, opts...)
if err != nil {
return nil, err
}
return &BatchUpdateUserLicensesOperation{
lro: longrunning.InternalNewOperation(*c.LROClient, resp),
}, nil
}

func (c *userLicenseGRPCClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```



```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx,
hds...)
opts =
append((*c.CallOptions).CancelOperation[0:len((*c.CallOptions).CancelOperation):len((*c.CallOptions).CancelOp
eration)], opts...)
err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 _, err = executeRPC(ctx, c.operationsClient.CancelOperation, req, settings.GRPC, c.logger, "CancelOperation")
 return err
}, opts...)
return err
}

```

```

func (c *userLicenseGRPCClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption) (*longrunningpb.Operation, error) {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
var resp *longrunningpb.Operation
err := gax.Invoke(ctx, func(ctx context.Context,
settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.GetOperation, req, settings.GRPC, c.logger, "GetOperation")
 return err
}, opts...)
if err != nil {
 return nil, err
}
return resp, nil
}

```

```

func (c *userLicenseGRPCClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,
opts ...gax.CallOption) *OperationIterator {
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

```

```

hds = append(c.xGoogHeaders, hds...)
ctx = gax.InsertMetadataIntoOutgoingContext(ctx, hds...)
opts =
append((*c.CallOptions).ListOperations[0:len((*c.CallOptions).ListOperations):len((*c.CallOptions).ListOperations
)], opts...)
it := &OperationIterator{ }
req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)

```

```

it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{}
 if pageToken != "" {
 req.PageToken = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 err := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 var err error
 resp, err = executeRPC(ctx, c.operationsClient.ListOperations, req, settings.GRPC, c.logger, "ListOperations")
 return err
 }, opts...)
 if err != nil {
 return nil, "", err
 }

 it.Response = resp
 return resp.GetOperations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// ListUserLicenses lists the
// User Licenses.
func (c *userLicenseRESTClient) ListUserLicenses(ctx context.Context, req
*discoveryenginepb.ListUserLicensesRequest, opts ...gax.CallOption) *UserLicenseIterator {
 it := &UserLicenseIterator{}
 req = proto.Clone(req).(*discoveryenginepb.ListUserLicensesRequest)
 unmarshalOptions := &protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*discoveryenginepb.UserLicense, string, error) {
 resp := &discoveryenginepb.ListUserLicensesResponse{}

```

```

if pageToken != "" {
 req.PageToken = pageToken
}
if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
} else if pageSize != 0 {
 req.PageSize = int32(pageSize)
}
baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return nil, "", err
}
baseUrl.Path += fmt.Sprintf("/v1/%v/userLicenses", req.GetParent())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")
if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v",
req.GetFilter()))
}
if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
}
if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListUserLicenses")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

```

```

 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response
= resp
return resp.GetUserLicenses(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// BatchUpdateUserLicenses updates the User License.
// This method is used for batch assign/unassign licenses to users.
func (c *userLicenseRESTClient) BatchUpdateUserLicenses(ctx context.Context, req
*discoveryenginepb.BatchUpdateUserLicensesRequest, opts ...gax.CallOption)
(*BatchUpdateUserLicensesOperation, error) {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}
 jsonReq, err := m.Marshal(req)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if
err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v:batchUpdateUserLicenses", req.GetParent())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")

```

```

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "parent", url.QueryEscape(req.GetParent()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err
:= executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "BatchUpdateUserLicenses")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}

override := fmt.Sprintf("/v1/%s", resp.GetName())
return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, resp),
 pollPath: override,
}, nil
}

// CancelOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) CancelOperation(ctx context.Context, req
*longrunningpb.CancelOperationRequest, opts ...gax.CallOption) error {
 m := protojson.MarshalOptions{AllowPartial: true, UseEnumNumbers: true}

```

```

jsonReq, err := m.Marshal(req)
if err != nil {
 return err
}

baseUrl, err := url.Parse(c.endpoint)
if err != nil {
 return err
}
baseUrl.Path += fmt.Sprintf("/v1/%v:cancel", req.GetName())

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery
= params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
return gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 _, err = executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "CancelOperation")
 return err
}, opts...)
}

// GetOperation is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) GetOperation(ctx context.Context, req *longrunningpb.GetOperationRequest,
opts ...gax.CallOption)
(*longrunningpb.Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v", req.GetName())

```

```

params := url.Values{}
params.Add("$alt", "json;enum-encoding=int")

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "name", url.QueryEscape(req.GetName()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts =
append((*c.CallOptions).GetOperation[0:len((*c.CallOptions).GetOperation):len((*c.CallOptions).GetOperation)],
opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &longrunningpb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET",
baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetOperation")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// ListOperations is a utility method from google.longrunning.Operations.
func (c *userLicenseRESTClient) ListOperations(ctx context.Context, req *longrunningpb.ListOperationsRequest,

```

```

opts ...gax.CallOption) *OperationIterator {
it := &OperationIterator{}
req = proto.Clone(req).(*longrunningpb.ListOperationsRequest)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
it.InternalFetch = func(pageSize int, pageToken string) ([]*longrunningpb.Operation, string, error) {
 resp := &longrunningpb.ListOperationsResponse{}
 if pageToken != "" {
 req.PageToken
 = pageToken
 }
 if pageSize > math.MaxInt32 {
 req.PageSize = math.MaxInt32
 } else if pageSize != 0 {
 req.PageSize = int32(pageSize)
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/v1/%v/operations", req.GetName())

 params := url.Values{}
 params.Add("$alt", "json;enum-encoding=int")
 if req.GetFilter() != "" {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req.GetPageSize() != 0 {
 params.Add("pageSize", fmt.Sprintf("%v", req.GetPageSize()))
 }
 if req.GetPageToken() != "" {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := append(c.xGoogHeaders, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path
 = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq.Header = headers

```



```

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "ListOperations")
 if err != nil {
 return err
 }
 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetOperations(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetPageSize())
it.pageInfo.Token = req.GetPageToken()

return it
}

// BatchUpdateUserLicensesOperation returns a new
// BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.
func (c *userLicenseGRPCClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 }
}

// BatchUpdateUserLicensesOperation returns a new BatchUpdateUserLicensesOperation from a given name.
// The name must be that of a previously created BatchUpdateUserLicensesOperation, possibly from a different
// process.

```

```

func (c *userLicenseRESTClient) BatchUpdateUserLicensesOperation(name string)
*BatchUpdateUserLicensesOperation {
 override := fmt.Sprintf("/v1/%s", name)
 return &BatchUpdateUserLicensesOperation{
 lro: longrunning.InternalNewOperation(*c.LROClient, &longrunningpb.Operation{Name: name}),
 pollPath: override,
 }
}

// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute

import (
 "bytes"
 "context"
 "fmt"
 "log/slog"
 "math"
 "net/http"
 "net/url"
 "time"

 computepb "cloud.google.com/go/compute/apiv1beta/compute"
 gax "github.com/googleapis/gax-go/v2"
 "google.golang.org/api/iterator"
 "google.golang.org/api/option"
 "google.golang.org/api/option/internaloption"
 httptransport "google.golang.org/api/transport/http"
 "google.golang.org/grpc"
 "google.golang.org/protobuf/encoding/protojson"
 "google.golang.org/protobuf/proto"
)

var

```

```
newLicensesClientHook clientHook
```

```
// LicensesCallOptions contains the retry settings for each method of LicensesClient.
```

```
type LicensesCallOptions struct {
 Delete []gax.CallOption
 Get []gax.CallOption
 GetIamPolicy []gax.CallOption
 Insert []gax.CallOption
 List []gax.CallOption
 SetIamPolicy []gax.CallOption
 TestIamPermissions []gax.CallOption
 Update []gax.CallOption
}
```

```
func defaultLicensesRESTCallOptions() *LicensesCallOptions {
```

```
 return &LicensesCallOptions{
 Delete: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 Get: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier:
1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 GetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
 },
 Insert: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
 },
 List: []gax.CallOption{
```

```

gax.WithTimeout(600000 * time.Millisecond),
gax.WithRetry(func() gax.Retryer {
 return gax.OnHTTPCodes(gax.Backoff{
 Initial: 100 * time.Millisecond,
 Max: 60000 * time.Millisecond,
 Multiplier: 1.30,
 },
 http.StatusGatewayTimeout,
 http.StatusServiceUnavailable)
 }),
},
SetIamPolicy: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
TestIamPermissions: []gax.CallOption{
 gax.WithTimeout(600000
* time.Millisecond),
},
Update: []gax.CallOption{
 gax.WithTimeout(600000 * time.Millisecond),
},
}
}

// internalLicensesClient is an interface that defines the methods available from Google Compute Engine API.
type internalLicensesClient interface {
 Close() error
 setGoogleClientInfo(...string)
 Connection() *grpc.ClientConn
 Delete(context.Context, *computepb.DeleteLicenseRequest, ...gax.CallOption) (*Operation, error)
 Get(context.Context, *computepb.GetLicenseRequest, ...gax.CallOption) (*computepb.License, error)
 GetIamPolicy(context.Context, *computepb.GetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 Insert(context.Context, *computepb.InsertLicenseRequest, ...gax.CallOption) (*Operation, error)
 List(context.Context, *computepb.ListLicensesRequest, ...gax.CallOption) *LicenseIterator
 SetIamPolicy(context.Context, *computepb.SetIamPolicyLicenseRequest, ...gax.CallOption) (*computepb.Policy,
 error)
 TestIamPermissions(context.Context,
 *computepb.TestIamPermissionsLicenseRequest, ...gax.CallOption) (*computepb.TestPermissionsResponse, error)
 Update(context.Context, *computepb.UpdateLicenseRequest, ...gax.CallOption) (*Operation, error)
}

// LicensesClient is a client for interacting with Google Compute Engine API.
// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
// method calls.
//
// The Licenses API.
type LicensesClient struct {

```

```

// The internal transport-dependent client.
internalClient internalLicensesClient

// The call options for this service.
CallOptions *LicensesCallOptions
}

// Wrapper methods routed to the internal client.

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *LicensesClient) Close() error {
 return c.internalClient.Close()
}

// setGoogleClientInfo sets the name and version of the application in
// the `x-goog-api-client` header
// passed on each request. Intended for
// use by Google-written clients.
func (c *LicensesClient) setGoogleClientInfo(keyval ...string) {
 c.internalClient.setGoogleClientInfo(keyval...)
}

// Connection returns a connection to the API service.
//
// Deprecated: Connections are now pooled so this method does not always
// return the same resource.
func (c *LicensesClient) Connection() *grpc.ClientConn {
 return c.internalClient.Connection()
}

// Delete deletes the specified license. Caution This resource is intended for use only by third-party partners who are
// creating Cloud Marketplace images.
func (c *LicensesClient) Delete(ctx context.Context, req *computepb.DeleteLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Delete(ctx, req, opts...)
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating Cloud Marketplace images.
func (c *LicensesClient) Get(ctx context.Context,
 req *computepb.GetLicenseRequest, opts ...gax.CallOption) (*computepb.License, error) {
 return c.internalClient.Get(ctx, req, opts...)
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest, opts

```

```

...gax.CallOption) (*computepb.Policy, error) {
 return c.internalClient.GetIamPolicy(ctx, req, opts...)
}

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts ...gax.CallOption)
(*Operation, error) {
 return c.internalClient.Insert(ctx, req, opts...)
}

// List retrieves the list of licenses available
in the specified project. This method does not get any licenses that belong to other projects, including licenses
attached to publicly-available images, like Debian 9. If you want to get a list of publicly-available licenses, use this
method to make a request to the respective image project, such as debian-cloud or windows-cloud. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts ...gax.CallOption)
*LicenseIterator {
 return c.internalClient.List(ctx, req, opts...)
}

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This
resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest, opts
...gax.CallOption) (*computepb.Policy, error) {
 return
 c.internalClient.SetIamPolicy(ctx, req, opts...)
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) TestIamPermissions(ctx context.Context, req
*computepb.TestIamPermissionsLicenseRequest, opts ...gax.CallOption) (*computepb.TestPermissionsResponse,
error) {
 return c.internalClient.TestIamPermissions(ctx, req, opts...)
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *LicensesClient) Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 return c.internalClient.Update(ctx, req, opts...)
}

// Methods, except Close, may be called concurrently. However, fields must not be modified concurrently with
method calls.
type

```

```

licensesRESTClient struct {
// The http endpoint to connect to.
endpoint string

// The http client.
httpClient *http.Client

// operationClient is used to call the operation-specific management service.
operationClient *GlobalOperationsClient

// The x-goog-* headers to be sent with each request.
xGoogHeaders []string

// Points back to the CallOptions field of the containing LicensesClient
CallOptions **LicensesCallOptions

logger *slog.Logger
}

// NewLicensesRESTClient creates a new licenses rest client.
//
// The Licenses API.
func NewLicensesRESTClient(ctx context.Context, opts ...option.ClientOption) (*LicensesClient, error) {
 clientOpts := append(defaultLicensesRESTClientOptions(), opts...)
 httpClient, endpoint, err := httptransport.NewClient(ctx, clientOpts...)
 if err != nil {
 return nil, err
 }

 callOpts := defaultLicensesRESTCallOptions()
 c := &licensesRESTClient{
 endpoint: endpoint,
 httpClient: httpClient,
 CallOptions: &callOpts,
 logger:
 internaloption.GetLogger(opts),
 }
 c.setGoogleClientInfo()

 o := []option.ClientOption{
 option.WithHTTPClient(httpClient),
 option.WithEndpoint(endpoint),
 }
 opC, err := NewGlobalOperationsRESTClient(ctx, o...)
 if err != nil {
 return nil, err
 }
 c.operationClient = opC

```

```

return &LicensesClient{internalClient: c, CallOptions: callOpts}, nil
}

func defaultLicensesRESTClientOptions() []option.ClientOption {
return []option.ClientOption{
 internaloption.WithDefaultEndpoint("https://compute.googleapis.com"),
 internaloption.WithDefaultEndpointTemplate("https://compute.UNIVERSE_DOMAIN"),
 internaloption.WithDefaultMTLSEndpoint("https://compute.mtls.googleapis.com"),
 internaloption.WithDefaultUniverseDomain("googleapis.com"),
 internaloption.WithDefaultAudience("https://compute.googleapis.com/"),
 internaloption.WithDefaultScopes(DefaultAuthScopes()...),
 internaloption.EnableNewAuthLibrary(),
}
}

// setGoogleClientInfo sets the name and version of the application
in
// the `x-goog-api-client` header passed on each request. Intended for
// use by Google-written clients.
func (c *licensesRESTClient) setGoogleClientInfo(keyval ...string) {
kv := append([]string{"gl-go", gax.GoVersion}, keyval...)
kv = append(kv, "gapic", getVersionClient(), "gax", gax.Version, "rest", "UNKNOWN", "pb", protoVersion)
c.xGoogHeaders = []string{
 "x-goog-api-client", gax.XGoogHeader(kv...),
}
}

// Close closes the connection to the API service. The user should invoke this when
// the client is no longer required.
func (c *licensesRESTClient) Close() error {
// Replace httpClient with nil to force cleanup.
c.httpClient = nil
if err := c.operationClient.Close(); err != nil {
 return err
}
return nil
}

// Connection returns a connection to the API service.
//
// Deprecated: This method always returns nil.
func (c *licensesRESTClient) Connection() *grpc.ClientConn {
return nil
}

// Delete deletes the specified license. Caution This resource is intended for

```



```

use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) Delete(ctx context.Context, req *compute.DeleteLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Delete[0:len((*c.CallOptions).Delete):len((*c.CallOptions).Delete)],
opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &compute.Operation{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("DELETE", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Delete")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
 })

```

```

 }, opts...)
 if e != nil {
 return nil, e
 }
 op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
 }
 return op, nil
}

// Get returns the specified License resource. Caution This resource is intended for use only by third-party partners
// who are creating
// Cloud Marketplace images.
func (c *licensesRESTClient) Get(ctx context.Context, req *computepb.GetLicenseRequest, opts ...gax.CallOption)
(*computepb.License, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts = append((*c.CallOptions).Get[0:len((*c.CallOptions).Get):len((*c.CallOptions).Get)], opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.License{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings)
 error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "Get")

```

```

if err != nil {
 return err
}

if err := unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// GetIamPolicy gets the access control policy for a resource. May be empty if no such policy or resource exists.
// Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) GetIamPolicy(ctx context.Context, req *computepb.GetIamPolicyLicenseRequest,
 opts ...gax.CallOption) (*computepb.Policy, error) {
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/getIamPolicy",
 req.GetProject(), req.GetResource())

 params := url.Values{}
 if req != nil && req.OptionsRequestedPolicyVersion != nil {
 params.Add("optionsRequestedPolicyVersion", fmt.Sprintf("%v", req.GetOptionsRequestedPolicyVersion()))
 }

 baseUrl.RawQuery = params.Encode()

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).GetIamPolicy[0:len((*c.CallOptions).GetIamPolicy):len((*c.CallOptions).GetIamPolicy)],
 opts...)
 unmarshal := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.Policy{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error

```

```

{
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "GetIamPolicy")
 if err != nil {
 return err
 }

 if err := unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

```

// Insert create a License resource in the specified project. Caution This resource is intended for use only by third-party partners who are creating Cloud Marketplace images.

```

func (c *licensesRESTClient) Insert(ctx context.Context, req *computepb.InsertLicenseRequest, opts
...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

```

```

baseUrl, err
:= url.Parse(c.endpoint)
if err != nil {
 return nil, err
}
baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

```

```

params := url.Values{}
if req != nil && req.RequestId != nil {

```

```

params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
}

baseUrl.RawQuery = params.Encode()

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v", "project", url.QueryEscape(req.GetProject()))}

hds = append(c.xGoogHeaders, hds...)
hds = append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Insert[0:len((*c.CallOptions).Insert):len((*c.CallOptions).Insert)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &computepb.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq,
err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Insert")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project: req.GetProject(),
 },
}
return op, nil

```

```

}

// List retrieves the list of licenses available in the specified project. This method does not get any licenses that
// belong to other projects, including licenses attached to publicly-available images, like Debian 9. If you want to get a
// list of publicly-available licenses, use this method to make a request to the respective image project, such as debian-
// cloud or windows-cloud. Caution This resource is intended
// for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient) List(ctx context.Context, req *computepb.ListLicensesRequest, opts
...gax.CallOption) *LicenseIterator {
 it := &LicenseIterator{}
 req = proto.Clone(req).(*computepb.ListLicensesRequest)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 it.InternalFetch = func(pageSize int, pageToken string) ([]*computepb.License, string, error) {
 resp := &computepb.LicensesListResponse{}
 if pageToken != "" {
 req.PageToken = proto.String(pageToken)
 }
 if pageSize > math.MaxInt32 {
 req.MaxResults = proto.Uint32(uint32(math.MaxInt32))
 } else if pageSize != 0 {
 req.MaxResults = proto.Uint32(uint32(pageSize))
 }
 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, "", err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses", req.GetProject())

 params := url.Values{}
 if req != nil && req.Filter != nil {
 params.Add("filter", fmt.Sprintf("%v", req.GetFilter()))
 }
 if req != nil && req.MaxResults != nil {
 params.Add("maxResults", fmt.Sprintf("%v", req.GetMaxResults()))
 }
 if req != nil && req.OrderBy != nil {
 params.Add("orderBy", fmt.Sprintf("%v", req.GetOrderBy()))
 }
 if req != nil && req.PageToken != nil {
 params.Add("pageToken", fmt.Sprintf("%v", req.GetPageToken()))
 }
 if req != nil && req.ReturnPartialSuccess != nil {
 params.Add("returnPartialSuccess", fmt.Sprintf("%v", req.GetReturnPartialSuccess()))
 }

 baseUrl.RawQuery = params.Encode()

```

```

// Build HTTP headers from client and context metadata.
hds := append(c.xGoogHeaders, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("GET", baseUrl.String(), nil)
 if err != nil {
 return
 }
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, nil, "List")
 if err != nil {
 return err
 }
 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, "", e
}
it.Response = resp
return resp.GetItems(), resp.GetNextPageToken(), nil
}

fetch := func(pageSize int, pageToken string) (string, error) {
 items, nextPageToken, err := it.InternalFetch(pageSize, pageToken)
 if err != nil {
 return "", err
 }
 it.items = append(it.items, items...)
 return nextPageToken, nil
}

it.pageInfo, it.nextFunc = iterator.NewPageInfo(fetch, it.bufLen, it.takeBuf)
it.pageInfo.MaxSize = int(req.GetMaxResults())
it.pageInfo.Token = req.GetPageToken()

return it
}

```

// SetIamPolicy sets the access control policy on the specified resource. Replaces any existing policy. Caution This

resource is intended for use only by third-party partners who are creating

Cloud Marketplace images.

```
func (c *licensesRESTClient) SetIamPolicy(ctx context.Context, req *computepb.SetIamPolicyLicenseRequest,
opts ...gax.CallOption) (*computepb.Policy, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetGlobalSetPolicyRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }
```

```
 baseUrl, err := url.Parse(c.endpoint)
```

```
 if err != nil {
 return nil, err
 }
```

```
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/setIamPolicy", req.GetProject(),
req.GetResource())
```

```
 // Build HTTP headers from client and context metadata.
```

```
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}
```

```
 hds = append(c.xGoogHeaders, hds...)
```

```
 hds = append(hds, "Content-Type", "application/json")
```

```
 headers := gax.BuildHeaders(ctx, hds...)
```

```
 opts =
```

```
append((*c.CallOptions).SetIamPolicy[0:len((*c.CallOptions).SetIamPolicy):len((*c.CallOptions).SetIamPolicy)],
opts...)
```

```
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
```

```
 resp := &computepb.Policy{}
```

```
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
```

```
 if settings.Path != "" {
```

```
 baseUrl.Path = settings.Path
```

```
 }
```

```
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
```

```
 if err != nil {
```

```
 return err
```

```
 }
```

```
 httpReq = httpReq.WithContext(ctx)
```

```
 httpReq.Header = headers
```

```
 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "SetIamPolicy")
```

```
 if err != nil {
```

```
 return err
```

```
 }
```

```
 if err := unmarshal(buf, resp); err != nil {
```

```
 return err
```



```

 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// TestIamPermissions returns permissions that a caller has on the specified resource. Caution This resource is
// intended for use only by third-party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
TestIamPermissions(ctx context.Context, req *computepb.TestIamPermissionsLicenseRequest, opts
...gax.CallOption) (*computepb.TestPermissionsResponse, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetTestPermissionsRequestResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v/testIamPermissions", req.GetProject(),
 req.GetResource())

 // Build HTTP headers from client and context metadata.
 hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
 url.QueryEscape(req.GetProject()), "resource", url.QueryEscape(req.GetResource()))}

 hds = append(c.xGoogHeaders, hds...)
 hds = append(hds, "Content-Type", "application/json")
 headers := gax.BuildHeaders(ctx, hds...)
 opts =
 append((*c.CallOptions).TestIamPermissions[0:len((*c.CallOptions).TestIamPermissions):len((*c.CallOptions).Tes
 tIamPermissions)],
 opts...)
 unmarshalOptions := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
 resp := &computepb.TestPermissionsResponse{}
 e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("POST", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {

```

```

 return err
}
httpReq = httpReq.WithContext(ctx)
httpReq.Header = headers

buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "TestIamPermissions")
if err != nil {
 return err
}

if err := unm.Unmarshal(buf, resp); err != nil {
 return err
}

return nil
}, opts...)
if e != nil {
 return nil, e
}
return resp, nil
}

// Update updates a License resource in the specified project. Caution This resource is intended for use only by third-
party partners who are creating Cloud Marketplace images.
func (c *licensesRESTClient)
Update(ctx context.Context, req *computepb.UpdateLicenseRequest, opts ...gax.CallOption) (*Operation, error) {
 m := protojson.MarshalOptions{AllowPartial: true}
 body := req.GetLicenseResource()
 jsonReq, err := m.Marshal(body)
 if err != nil {
 return nil, err
 }

 baseUrl, err := url.Parse(c.endpoint)
 if err != nil {
 return nil, err
 }
 baseUrl.Path += fmt.Sprintf("/compute/beta/projects/%v/global/licenses/%v", req.GetProject(), req.GetLicense())

 params := url.Values{}
 if req != nil && req.RequestId != nil {
 params.Add("requestId", fmt.Sprintf("%v", req.GetRequestId()))
 }
 if req != nil && req.UpdateMask != nil {
 params.Add("updateMask", fmt.Sprintf("%v", req.GetUpdateMask()))
 }

 baseUrl.RawQuery = params.Encode()

```

```

// Build HTTP headers from client and context metadata.
hds := []string{"x-goog-request-params", fmt.Sprintf("%s=%v&%s=%v", "project",
url.QueryEscape(req.GetProject()), "license", url.QueryEscape(req.GetLicense()))}

hds = append(c.xGoogHeaders, hds...)
hds
= append(hds, "Content-Type", "application/json")
headers := gax.BuildHeaders(ctx, hds...)
opts = append((*c.CallOptions).Update[0:len((*c.CallOptions).Update):len((*c.CallOptions).Update)], opts...)
unm := protojson.UnmarshalOptions{AllowPartial: true, DiscardUnknown: true}
resp := &compute.Operation{}
e := gax.Invoke(ctx, func(ctx context.Context, settings gax.CallSettings) error {
 if settings.Path != "" {
 baseUrl.Path = settings.Path
 }
 httpReq, err := http.NewRequest("PATCH", baseUrl.String(), bytes.NewReader(jsonReq))
 if err != nil {
 return err
 }
 httpReq = httpReq.WithContext(ctx)
 httpReq.Header = headers

 buf, err := executeHTTPRequest(ctx, c.httpClient, httpReq, c.logger, jsonReq, "Update")
 if err != nil {
 return err
 }

 if err := unm.Unmarshal(buf, resp); err != nil {
 return err
 }

 return nil
}, opts...)
if e != nil {
 return nil, e
}
op := &Operation{
 &globalOperationsHandle{
 c: c.operationClient,
 proto: resp,
 project:
req.GetProject(),
 },
}
return op, nil
}

```

```
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go. DO NOT EDIT.
// versions:
// protoc-gen-go v1.35.2
// protoc v4.25.7
// source: google/ai/generativelanguage/v1beta/permission.proto

package generativelanguagepb

import (
 _ "google.golang.org/genproto/googleapis/api/annotations"
 protoreflect "google.golang.org/protobuf/reflect/protoreflect"
 protoimpl "google.golang.org/protobuf/runtime/protoimpl"
 reflect "reflect"
 sync
 "sync"
)

const (
 // Verify that this generated code is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(20 - protoimpl.MinVersion)
 // Verify that runtime/protoimpl is sufficiently up-to-date.
 _ = protoimpl.EnforceVersion(protoimpl.MaxVersion - 20)
)

// Defines types of the grantee of this permission.
type Permission_GranteeType int32

const (
 // The default value. This value is unused.
 Permission_GRANTEE_TYPE_UNSPECIFIED Permission_GranteeType = 0
 // Represents a user. When set, you must provide email_address for the user.
 Permission_USER Permission_GranteeType = 1
 // Represents a group. When set, you must provide email_address for the
 // group.

```

```

Permission_GROUP Permission_GrantType = 2
// Represents access to everyone. No extra information is required.
Permission_EVERYONE Permission_GrantType = 3
)

// Enum value maps for Permission_GrantType.
var (
 Permission_GrantType_name = map[int32]string{
 0: "GRANTEE_TYPE_UNSPECIFIED",
 1: "USER",
 2:
 "GROUP",
 3: "EVERYONE",
 }
 Permission_GrantType_value = map[string]int32{
 "GRANTEE_TYPE_UNSPECIFIED": 0,
 "USER": 1,
 "GROUP": 2,
 "EVERYONE": 3,
 }
)

func (x Permission_GrantType) Enum() *Permission_GrantType {
 p := new(Permission_GrantType)
 *p = x
 return p
}

func (x Permission_GrantType) String() string {
 return protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_GrantType) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0].Descriptor()
}

func (Permission_GrantType) Type() protoreflect.EnumType {
 return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[0]
}

func (x Permission_GrantType) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_GrantType.Descriptor instead.
func (Permission_GrantType) EnumDescriptor() ([]byte,
[]int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 0}
}

```

```

}

// Defines the role granted by this permission.
type Permission_Role int32

const (
 // The default value. This value is unused.
 Permission_ROLE_UNSPECIFIED Permission_Role = 0
 // Owner can use, update, share and delete the resource.
 Permission_OWNER Permission_Role = 1
 // Writer can use, update and share the resource.
 Permission_WRITER Permission_Role = 2
 // Reader can use the resource.
 Permission_READER Permission_Role = 3
)

// Enum value maps for Permission_Role.
var (
 Permission_Role_name = map[int32]string{
 0: "ROLE_UNSPECIFIED",
 1: "OWNER",
 2: "WRITER",
 3: "READER",
 }
 Permission_Role_value = map[string]int32{
 "ROLE_UNSPECIFIED": 0,
 "OWNER": 1,
 "WRITER": 2,
 "READER": 3,
 }
)

func (x Permission_Role) Enum() *Permission_Role {
 p := new(Permission_Role)
 *p = x
 return p
}

func (x Permission_Role) String() string {
 return
 protoimpl.X.EnumStringOf(x.Descriptor(), protoreflect.EnumNumber(x))
}

func (Permission_Role) Descriptor() protoreflect.EnumDescriptor {
 return file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1].Descriptor()
}

func (Permission_Role) Type() protoreflect.EnumType {

```

```

return &file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes[1]
}

func (x Permission_Role) Number() protoreflect.EnumNumber {
 return protoreflect.EnumNumber(x)
}

// Deprecated: Use Permission_Role.Descriptor instead.
func (Permission_Role) EnumDescriptor() ([]byte, []int) {
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0, 1}
}

// Permission resource grants user, group or the rest of the world access to the
// PaLM API resource (e.g. a tuned model, corpus).
//
// A role is a collection of permitted operations that allows users to perform
// specific actions on PaLM API resources. To make them available to users,
// groups, or service accounts,
// you assign roles. When you assign a role, you
// grant permissions that the role contains.
//
// There are three concentric roles. Each role is a superset of the previous
// role's permitted operations:
//
// - reader can use the resource (e.g. tuned model, corpus) for inference
// - writer has reader's permissions and additionally can edit and share
// - owner has writer's permissions and additionally can delete
type Permission struct {
 state protoimpl.MessageState
 sizeCache protoimpl.SizeCache
 unknownFields protoimpl.UnknownFields

 // Output only. Identifier. The permission name. A unique name will be
 // generated on create. Examples:
 //
 // tunedModels/{tuned_model}/permissions/{permission}
 // corpora/{corpus}/permissions/{permission}
 //
 // Output only.
 Name string `protobuf:"bytes,1,opt,name=name,proto3" json:"name,omitempty"`
 // Optional. Immutable. The type of the grantee.
 GranteeType *Permission_GranteeType
 `protobuf:"varint,2,opt,name=grantee_type,json=granteeType,proto3,enum=google.ai.generativelanguage.v1beta.Permission_GranteeType,oneof"
 json:"grantee_type,omitempty"`
 // Optional. Immutable. The email address of the user or group which this
 // permission refers. Field is not set when permission's grantee type is
 // EVERYONE.

```

```

EmailAddress *string `protobuf:"bytes,3,opt,name=email_address,json=emailAddress,proto3,oneof"
json:"email_address,omitempty"`
// Required. The role granted by this permission.
Role *Permission_Role
`protobuf:"varint,4,opt,name=role,proto3,enum=google.ai.generativelanguage.v1beta.Permission_Role,oneof"
json:"role,omitempty"`
}

func (x *Permission) Reset() {
*x = Permission{}
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
ms.StoreMessageInfo(mi)
}

func (x *Permission) String() string {
return protoimpl.X.MessageStringOf(x)
}

func (*Permission) ProtoMessage() {}

func (x *Permission) ProtoReflect() protoreflect.Message
{
mi := &file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0]
if x != nil {
ms := protoimpl.X.MessageStateOf(protoimpl.Pointer(x))
if ms.LoadMessageInfo() == nil {
ms.StoreMessageInfo(mi)
}
return ms
}
return mi.MessageOf(x)
}

// Deprecated: Use Permission.ProtoReflect.Descriptor instead.
func (*Permission) Descriptor() ([]byte, []int) {
return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP(), []int{0}
}

func (x *Permission) GetName() string {
if x != nil {
return x.Name
}
return ""
}

func (x *Permission) GetGranteeType() Permission_GranteeType {
if x != nil && x.GranteeType != nil {

```



```

 return *x.GranteeType
}
return Permission_GRANTEE_TYPE_UNSPECIFIED
}

func (x *Permission) GetEmailAddress() string {
 if x != nil && x.EmailAddress != nil {
 return *x.EmailAddress
 }
 return ""
}

func (x *Permission) GetRole() Permission_Role {
 if x != nil && x.Role != nil {
 return *x.Role
 }
 return Permission_ROLE_UNSPECIFIED
}

var
File_google_ai_generativelanguage_v1beta_permission_proto protoreflect.FileDescriptor

var file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = []byte{
 0x0a, 0x34, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65,
 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x76,
 0x31, 0x62, 0x65, 0x74, 0x61, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x12, 0x23, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x1a, 0x1f, 0x67, 0x6f, 0x6f,
 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x2f, 0x66, 0x69, 0x65, 0x6c, 0x64, 0x5f, 0x62, 0x65,
 0x68, 0x61, 0x76, 0x69, 0x6f, 0x72, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x1a, 0x19, 0x67, 0x6f,
 0x6f, 0x67, 0x6c, 0x65, 0x2f, 0x61, 0x70, 0x69,
 0x2f, 0x72, 0x65, 0x73, 0x6f, 0x75, 0x72, 0x63,
 0x65, 0x2e, 0x70, 0x72, 0x6f, 0x74, 0x6f, 0x22, 0x85, 0x05, 0x0a, 0x0a, 0x50, 0x65, 0x72, 0x6d,
 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x12, 0x1a, 0x0a, 0x04, 0x6e, 0x61, 0x6d, 0x65, 0x18, 0x01,
 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x03, 0xe0, 0x41, 0x08, 0x52, 0x04, 0x6e, 0x61,
 0x6d, 0x65, 0x12, 0x6b, 0x0a, 0x0c, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f, 0x74, 0x79,
 0x70, 0x65, 0x18, 0x02, 0x20, 0x01, 0x28, 0x0e, 0x32, 0x3b, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c,
 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c,
 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50,
 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x2e, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65,
 0x65, 0x54, 0x79, 0x70, 0x65, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x00, 0x52,
 0x0b, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x54, 0x79, 0x70, 0x65, 0x88,
 0x01, 0x01, 0x12,
 0x30, 0x0a, 0x0d, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73,
 0x18, 0x03, 0x20, 0x01, 0x28, 0x09, 0x42, 0x06, 0xe0, 0x41, 0x01, 0xe0, 0x41, 0x05, 0x48, 0x01,
 0x52, 0x0c, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x41, 0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x88, 0x01,
 0x01, 0x12, 0x52, 0x0a, 0x04, 0x72, 0x6f, 0x6c, 0x65, 0x18, 0x04, 0x20, 0x01, 0x28, 0x0e, 0x32,

```

```

0x34, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61, 0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65,
0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2e, 0x76,
0x31, 0x62, 0x65, 0x74, 0x61, 0x2e, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e,
0x2e, 0x52, 0x6f, 0x6c, 0x65, 0x42, 0x03, 0xe0, 0x41, 0x02, 0x48, 0x02, 0x52, 0x04, 0x72, 0x6f,
0x6c, 0x65, 0x88, 0x01, 0x01, 0x22, 0x4e, 0x0a, 0x0b, 0x47, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65,
0x54, 0x79, 0x70, 0x65, 0x12, 0x1c, 0x0a, 0x18, 0x47, 0x52, 0x41, 0x4e, 0x54, 0x45, 0x45, 0x5f,
0x54, 0x59,
0x50, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45, 0x44,
0x10, 0x00, 0x12, 0x08, 0x0a, 0x04, 0x55, 0x53, 0x45, 0x52, 0x10, 0x01, 0x12, 0x09, 0x0a, 0x05,
0x47, 0x52, 0x4f, 0x55, 0x50, 0x10, 0x02, 0x12, 0x0c, 0x0a, 0x08, 0x45, 0x56, 0x45, 0x52, 0x59,
0x4f, 0x4e, 0x45, 0x10, 0x03, 0x22, 0x3f, 0x0a, 0x04, 0x52, 0x6f, 0x6c, 0x65, 0x12, 0x14, 0x0a,
0x10, 0x52, 0x4f, 0x4c, 0x45, 0x5f, 0x55, 0x4e, 0x53, 0x50, 0x45, 0x43, 0x49, 0x46, 0x49, 0x45,
0x44, 0x10, 0x00, 0x12, 0x09, 0x0a, 0x05, 0x4f, 0x57, 0x4e, 0x45, 0x52, 0x10, 0x01, 0x12, 0x0a,
0x0a, 0x06, 0x57, 0x52, 0x49, 0x54, 0x45, 0x52, 0x10, 0x02, 0x12, 0x0a, 0x0a, 0x06, 0x52, 0x45,
0x41, 0x44, 0x45, 0x52, 0x10, 0x03, 0x3a, 0xaa, 0x01, 0xea, 0x41, 0xa6, 0x01, 0x0a, 0x2c, 0x67,
0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67,
0x65, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x61, 0x70, 0x69, 0x73, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x50, 0x65, 0x72, 0x6d, 0x69, 0x73,
0x73, 0x69, 0x6f, 0x6e, 0x12, 0x32, 0x74, 0x75, 0x6e,
0x65, 0x64, 0x4d, 0x6f, 0x64, 0x65, 0x6c, 0x73, 0x2f, 0x7b, 0x74, 0x75, 0x6e, 0x65, 0x64, 0x5f,
0x6d, 0x6f, 0x64, 0x65, 0x6c, 0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f,
0x6e, 0x73, 0x2f, 0x7b, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x12,
0x29, 0x63, 0x6f, 0x72, 0x70, 0x6f, 0x72, 0x61, 0x2f, 0x7b, 0x63, 0x6f, 0x72, 0x70, 0x75, 0x73,
0x7d, 0x2f, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x2f, 0x7b, 0x70,
0x65, 0x72, 0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x7d, 0x2a, 0x0b, 0x70, 0x65, 0x72, 0x6d,
0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x73, 0x32, 0x0a, 0x70, 0x65, 0x72, 0x6d, 0x69, 0x73, 0x73,
0x69, 0x6f, 0x6e, 0x42, 0x0f, 0x0a, 0x0d, 0x5f, 0x67, 0x72, 0x61, 0x6e, 0x74, 0x65, 0x65, 0x5f,
0x74, 0x79, 0x70, 0x65, 0x42, 0x10, 0x0a, 0x0e, 0x5f, 0x65, 0x6d, 0x61, 0x69, 0x6c, 0x5f, 0x61,
0x64, 0x64, 0x72, 0x65, 0x73, 0x73, 0x42, 0x07, 0x0a, 0x05, 0x5f, 0x72,
0x6f, 0x6c, 0x65, 0x42,
0x9b, 0x01, 0x0a, 0x27, 0x63, 0x6f, 0x6d, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x61,
0x69, 0x2e, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67,
0x75, 0x61, 0x67, 0x65, 0x2e, 0x76, 0x31, 0x62, 0x65, 0x74, 0x61, 0x42, 0x0f, 0x50, 0x65, 0x72,
0x6d, 0x69, 0x73, 0x73, 0x69, 0x6f, 0x6e, 0x50, 0x72, 0x6f, 0x74, 0x6f, 0x50, 0x01, 0x5a, 0x5d,
0x63, 0x6c, 0x6f, 0x75, 0x64, 0x2e, 0x67, 0x6f, 0x6f, 0x67, 0x6c, 0x65, 0x2e, 0x63, 0x6f, 0x6d,
0x2f, 0x67, 0x6f, 0x2f, 0x61, 0x69, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76,
0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x2f, 0x61, 0x70, 0x69, 0x76, 0x31, 0x62,
0x65, 0x74, 0x61, 0x2f, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74, 0x69, 0x76, 0x65, 0x6c, 0x61,
0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x3b, 0x67, 0x65, 0x6e, 0x65, 0x72, 0x61, 0x74,
0x69, 0x76, 0x65, 0x6c, 0x61, 0x6e, 0x67, 0x75, 0x61, 0x67, 0x65, 0x70, 0x62, 0x62, 0x06, 0x70,
0x72,
0x6f, 0x74, 0x6f, 0x33,
}

```

```

var (
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce sync.Once
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc

```

```

)

func file_google_ai_generativelanguage_v1beta_permission_proto_rawDescGZIP() []byte {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescOnce.Do(func() {
 file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData =
protoimpl.X.CompressGZIP(file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData)
 })
 return file_google_ai_generativelanguage_v1beta_permission_proto_rawDescData
}

var file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes = make([]protoimpl.EnumInfo, 2)
var file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes = make([]protoimpl.MessageInfo, 1)
var file_google_ai_generativelanguage_v1beta_permission_proto_goTypes = []any{
 (Permission_GrantType)(0),
 // 0: google.ai.generativelanguage.v1beta.Permission.GrantType
 (Permission_Role)(0), // 1: google.ai.generativelanguage.v1beta.Permission.Role
 (*Permission)(nil), // 2: google.ai.generativelanguage.v1beta.Permission
}

var file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = []int32{
 0, // 0: google.ai.generativelanguage.v1beta.Permission.grantee_type:type_name ->
google.ai.generativelanguage.v1beta.Permission.GrantType
 1, // 1: google.ai.generativelanguage.v1beta.Permission.role:type_name ->
google.ai.generativelanguage.v1beta.Permission.Role
 2, // [2:2] is the sub-list for method output_type
 2, // [2:2] is the sub-list for method input_type
 2, // [2:2] is the sub-list for extension type_name
 2, // [2:2] is the sub-list for extension extendee
 0, // [0:2] is the sub-list for field type_name
}

func init() { file_google_ai_generativelanguage_v1beta_permission_proto_init() }
func file_google_ai_generativelanguage_v1beta_permission_proto_init()
{
 if File_google_ai_generativelanguage_v1beta_permission_proto != nil {
 return
 }
 file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes[0].OneofWrappers = []any{}
 type x struct{}
 out := protoimpl.TypeBuilder{
 File: protoimpl.DescBuilder{
 GoPackagePath: reflect.TypeOf(x{}).PkgPath(),
 RawDescriptor: file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc,
 NumEnums: 2,
 NumMessages: 1,
 NumExtensions: 0,
 NumServices: 0,
 },
 GoTypes: file_google_ai_generativelanguage_v1beta_permission_proto_goTypes,
 }

```

```

DependencyIndexes: file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs,
EnumInfos: file_google_ai_generativelanguage_v1beta_permission_proto_enumTypes,
MessageInfos: file_google_ai_generativelanguage_v1beta_permission_proto_msgTypes,
}.Build()
File_google_ai_generativelanguage_v1beta_permission_proto = out.File
file_google_ai_generativelanguage_v1beta_permission_proto_rawDesc = nil
file_google_ai_generativelanguage_v1beta_permission_proto_goTypes
= nil
file_google_ai_generativelanguage_v1beta_permission_proto_depIdxs = nil
}
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

//go:build go1.23

package discoveryengine_test

import (
 "context"

 discoveryengine "cloud.google.com/go/discoveryengine/apiv1"
 discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"
 longrunningpb "cloud.google.com/go/longrunning/autogen/longrunningpb"
)

func ExampleUserLicenseClient_ListUserLicenses_all() {
 ctx := context.Background()
 //
 This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := discoveryengine.NewUserLicenseClient(ctx)
 if err != nil {

```

```

// TODO: Handle error.
}
defer c.Close()

req := &discoveryenginepb.ListUserLicensesRequest{
// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#ListUserLicensesRequest.
}
for resp, err := range c.ListUserLicenses(ctx, req).All() {
if err != nil {
// TODO: Handle error and break/return/continue. Iteration will stop after any error.
}
// TODO: Use resp.
_ = resp
}
}

func ExampleUserLicenseClient_ListOperations_all() {
ctx := context.Background()
//
This snippet has been automatically generated and should be regarded as a code template only.
// It will require modifications to work:
// - It may require correct/in-range values for request initialization.
// - It may require specifying regional endpoints when creating the service client as shown in:
// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
// TODO: Handle error.
}
defer c.Close()

req := &longrunningpb.ListOperationsRequest{
// TODO: Fill request struct fields.
// See https://pkg.go.dev/cloud.google.com/go/longrunning/autogen/longrunningpb#ListOperationsRequest.
}
for resp, err := range c.ListOperations(ctx, req).All() {
if err != nil {
// TODO: Handle error and break/return/continue. Iteration will stop after any error.
}
// TODO: Use resp.
_ = resp
}
}

```

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the



content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

// Copyright 2025 Google LLC

//

// Licensed under the Apache License, Version 2.0 (the "License");

// you may not use this file except in compliance with the License.

// You may obtain a copy of the License at

//

// <https://www.apache.org/licenses/LICENSE-2.0>

//

// Unless required by applicable law or agreed to in writing, software

// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

// See the License for the specific language governing permissions and

// limitations under the License.

// Code generated by protoc-gen-go\_gapic. DO NOT EDIT.

// [START discoveryengine\_v1\_generated\_UserLicenseService\_BatchUpdateUserLicenses\_sync]

package main

import (

    "context"

    discoveryengine "cloud.google.com/go/discoveryengine/apiv1"

    discoveryenginepb "cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb"

)

func main() {

    ctx := context.Background()

    // This snippet has been automatically generated  
    and should be regarded as a code template only.

    // It will require modifications to work:

    // - It may require correct/in-range values for request initialization.

    // - It may require specifying regional endpoints when creating the service client as shown in:

```

// https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
c, err := discoveryengine.NewUserLicenseClient(ctx)
if err != nil {
 // TODO: Handle error.
}
defer c.Close()

req := &discoveryenginepb.BatchUpdateUserLicensesRequest{
 // TODO: Fill request struct fields.
 // See
 https://pkg.go.dev/cloud.google.com/go/discoveryengine/apiv1/discoveryenginepb#BatchUpdateUserLicensesReque
 st.
}
op, err := c.BatchUpdateUserLicenses(ctx, req)
if err != nil {
 // TODO: Handle error.
}

resp, err := op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

// [END discoveryengine_v1_generated_UserLicenseService_BatchUpdateUserLicenses_sync]
// Copyright 2025 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// Code generated by protoc-gen-go_gapic. DO NOT EDIT.

package compute_test

import (
 "context"

```

```

compute "cloud.google.com/go/compute/apiv1"
computepb "cloud.google.com/go/compute/apiv1/compute_pb"
"google.golang.org/api/iterator"
)

```

```

func ExampleNewLicensesRESTClient() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will
 require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 // TODO: Use client.
 _ = c
}

```

```

func ExampleLicensesClient_Delete() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.DeleteLicenseRequest{
 //
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute_pb#DeleteLicenseRequest.
 }
 op, err := c.Delete(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {

```

```

// TODO: Handle error.
}
}

func ExampleLicensesClient_Get() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetLicenseRequest.
 }
 resp,
 err := c.Get(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_GetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.GetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#GetIamPolicyLicenseRequest.
 }
 resp, err := c.GetIamPolicy(ctx, req)

```

```

if err != nil {
 // TODO: Handle error.
}
// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_Insert()
{
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.InsertLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#InsertLicenseRequest.
 }
 op, err := c.Insert(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }

 err = op.Wait(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

func ExampleLicensesClient_List() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template
 // only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

defer c.Close()

req := &computepb.ListLicensesRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#ListLicensesRequest.
}
it := c.List(ctx, req)
for {
 resp, err := it.Next()
 if err == iterator.Done {
 break
 }
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp

 // If you need to access the underlying RPC response,
 // you can do so by casting the `Response` as below.
 // Otherwise, remove this line. Only populated after
 // first call to Next(). Not safe for concurrent access.
 _ = it.Response.(*computepb.LicensesListResponse)
}
}

func
ExampleLicensesClient_SetIamPolicy() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &computepb.SetIamPolicyLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/computepb#SetIamPolicyLicenseRequest.
 }
 resp, err := c.SetIamPolicy(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
}

```

```

// TODO: Use resp.
_ = resp
}

func ExampleLicensesClient_TestIamPermissions() {
 ctx := context.Background()
 // This snippet has been automatically
 generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.TestIamPermissionsLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#TestIamPermissionsLicenseRequest.
 }
 resp, err := c.TestIamPermissions(ctx, req)
 if err != nil {
 // TODO: Handle error.
 }
 // TODO: Use resp.
 _ = resp
}

func ExampleLicensesClient_Update() {
 ctx := context.Background()
 // This snippet has been automatically generated and should be regarded as a code template only.
 // It will require modifications to work:
 // - It may
 require correct/in-range values for request initialization.
 // - It may require specifying regional endpoints when creating the service client as shown in:
 // https://pkg.go.dev/cloud.google.com/go#hdr-Client_Options
 c, err := compute.NewLicensesRESTClient(ctx)
 if err != nil {
 // TODO: Handle error.
 }
 defer c.Close()

 req := &compute.UpdateLicenseRequest{
 // TODO: Fill request struct fields.
 // See https://pkg.go.dev/cloud.google.com/go/compute/apiv1/compute#UpdateLicenseRequest.
 }

```



```

op, err := c.Update(ctx, req)
if err != nil {
 // TODO: Handle error.
}

err = op.Wait(ctx)
if err != nil {
 // TODO: Handle error.
}
}

```

# 1.114 otel-metric 1.36.0

## 1.114.1 Available under license :

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,  
each Contributor hereby grants to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
(except as stated in this section) patent license to make, have made,  
use, offer to sell, sell, import, and otherwise transfer the Work,  
where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.115 otel-trace 1.36.0

## 1.115.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions



of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.116 bobesa-go-domain-util 0.0.0-20250410211237-17ab3b2f4a95

### 1.116.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Vlastimil enfeld

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.117 genproto-googleapis-rpc 0.0.0-20250603155806-513f23925822

## 1.117.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.118 github.com/bobesa/go-domain-util

## 0.0.0-20250410211237-17ab3b2f4a95

### 1.118.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Vlastimil enfeld

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.119 contrib-instrumentation-net-http-otelhttp 0.61.0

### 1.119.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.



"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.120 librdkafka 2.10.0-r0

### 1.120.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

### 1.120.2 Available under license :

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

```
/*
 * Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Sponsored in part by the Defense Advanced Research Projects
 * Agency (DARPA) and Air Force Research Laboratory,
 * Air Force
 * Materiel Command, USAF, under agreement number F39502-99-1-0512.
 */
/*-
 * Copyright (c) 2000 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by Dieter Baron and Thomas Klausner.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
```

- \* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer in the
- \* documentation and/or other materials provided with the distribution.
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
- \* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
- \* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
- AND FITNESS FOR A PARTICULAR
- \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
- \* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
- \* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
- \* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
- \* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
- \* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- \* POSSIBILITY OF SUCH DAMAGE.
- \*/

src/rdxxhash.[ch] src/lz4\*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this  
list of conditions and the following disclaimer in the documentation and/or  
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL,  
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

parts of src/rdfnv1a.c: [http://www.isthe.com/chongo/src/fnv/hash\\_32a.c](http://www.isthe.com/chongo/src/fnv/hash_32a.c)

Please do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> /oo/  
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

FindZstd.cmake: [git@github.com:facebook/folly](https://github.com/facebook/folly).git 87f1a403b49552dae75ae94c8610dd5979913477

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object



form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for

reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed

to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if

You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

# For src/crc32c.c copied (with modifications) from

# <http://stackoverflow.com/a/17646775/1821055>

/\* crc32c.c -- compute CRC-32C using the Intel crc32 instruction

\* Copyright (C) 2013 Mark Adler

\* Version 1.1 1 Aug 2013 Mark Adler

\*/

/\*

This software is provided 'as-is', without any express or implied warranty. In no event will the author be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler

madler@alumni.caltech.edu

\*/

For sys/queue.h:

\* Copyright (c) 1991, 1993

- \* The Regents of the University of California. All rights reserved.
- \*
- \* Redistribution and use in source and binary forms, with or without
- \* modification, are permitted provided that the following conditions
- \* are met:
- \* 1. Redistributions of source code must retain the above copyright
- \* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer in the
- \* documentation and/or other materials provided with the distribution.
- \* 4. Neither the name of the University nor the names of its contributors
- \* may be used to endorse or promote products derived from this software
- \* without specific prior written permission.
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- \* IMPLIED WARRANTIES
- OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- \* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- \* SUCH DAMAGE.
- \*
- \* @(#)queue.h 8.5 (Berkeley) 8/20/94
- \* \$FreeBSD\$

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
THE SOFTWARE.  
LICENSE

-----  
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill  
2023, Confluent Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,  
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimer in the documentation  
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE  
LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

-----  
For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished  
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

-----  
# For src/crc32c.c copied (with modifications) from  
# <http://stackoverflow.com/a/17646775/1821055>

/\* crc32c.c -- compute CRC-32C using the Intel crc32 instruction  
\* Copyright (C) 2013 Mark Adler  
\* Version 1.1 1 Aug 2013 Mark Adler  
\*/

/\*  
This software is provided 'as-is',  
without any express or implied  
warranty. In no event will the author be held liable for any damages  
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,  
including commercial applications, and to alter it and redistribute it  
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler  
madler@alumni.caltech.edu  
\*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: [http://www.isthe.com/chongo/src/fnv/hash\\_32a.c](http://www.isthe.com/chongo/src/fnv/hash_32a.c)

Please

do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> /oo/

<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

-----  
This license covers src/rdhdrhistogram.c which is a C port of  
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>  
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

-----  
The MIT License  
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH  
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
THE SOFTWARE

LICENSE.lz4

-----  
src/rdxxhash.[ch] src/lz4\*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this  
list of conditions and the following disclaimer in the documentation and/or  
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  
PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

-----  
parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

//-----

// MurmurHash2 was written by Austin Appleby, and is placed in the public  
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person



obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

LICENSE.nanopb

-----  
For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or  
implied warranty. In no event will the authors be held liable  
for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any  
purpose, including commercial applications, and to alter it and  
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you  
must not claim that you wrote the original software. If you use  
this software in a product, an acknowledgment in the product  
documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and  
must not be misrepresented as being the original software.
3. This notice may  
not be removed or altered from any source  
distribution.

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution

incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

LICENSE.pycrc

-----  
The following license applies to the files rdcrc32.c and rdcrc32.h which  
have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,  
free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.queue

-----  
For

sys/queue.h:

\* Copyright (c) 1991, 1993  
\* The Regents of the University of California. All rights reserved.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\* 4. Neither the name of the University nor the names of its contributors  
\* may be used to endorse or promote products derived from this software  
\* without specific prior written permission.  
\*  
\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND  
\* ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.  
\*  
\* @(#)queue.h 8.5 (Berkeley) 8/20/94  
\* \$FreeBSD\$

LICENSE.regex

-----  
regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

#### LICENSE.snappy

```

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
originally retrieved from http://github.com/andikleen/snappy-c #
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####
```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### LICENSE.tinycthread

```

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9
```



## License

-----

Copyright (c) 2012 Marcus Geelnard  
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

## LICENSE.wingetopt

-----

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

/\*

\* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

\*

\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
\* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR  
\* ANY DAMAGES

\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
\* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
\* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

\*

\* Sponsored in part by the Defense Advanced Research Projects

```

* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*_
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice,
* this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
#####
LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h
originally retrieved from http://github.com/andikleen/snappy-c
git revision 8015fd28739b9a6076ebaa6c53fe27bc238d219
#####

```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

\* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

.

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: \*

Copyright: 2012-2022, Magnus Edenhill; 2023, Confluent Inc.

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files: src/snappy.c src/snappy.h src/snappy\_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

.

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: src/rdxxhash.h src/rdxxhash.c

Copyright: 2012-2014, Yann Collet

License: BSD-2-clause

Files: src/regex.h src/regex.c

Copyright: n/a

License: public-domain

License: public-domain

The files tagged with this license contain the following paragraphs:

.

These libraries are in the public domain (or the equivalent where that is not

possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

Files: debian/\*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

BSD-style without advertising clause

Copyright (c) 2016-2019, Magnus Edenhill

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source

distribution.

parts of src/rdmurmur2.c: [git@github.com:abandoned/murmur2.git](https://github.com:abandoned/murmur2.git)

MurMurHash2 Library

//-----

// MurmurHash2 was written by Austin Appleby, and is placed in the public  
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

.

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: \*

Copyright: 2012-2022, Magnus Edenhill; 2023 Confluent Inc.

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files: src/snappy.c src/snappy.h src/snappy\_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

.

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.



\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.  
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: debian/\*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

This license covers src/rdhdrhistogram.c which is a C port of Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram> at revision 3a0bb77429bd3a61596f5e8a3172445844342120

-----  
The MIT License (MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

From <https://github.com/tinychthread/tinychthread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

-----

Copyright (c) 2012 Marcus Geelnard  
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

The following license applies to the files rdcrc32.c and rdcrc32.h which have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.121 [github.com/google/uuid](https://github.com/google/uuid) 1.6.0

## 1.121.1 Available under license :

Paul Borman <borman@google.com>

bmatsuo

shawnps

theory

jboverfelt

dsymonds

cd1

wallclockbuilder

dansouza

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.122 cJSON 1.7.18-r1

## 1.122.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit  
(<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

## 1.122.2 Available under license :

Contributors

=====

Original Author:

- [Dave Gamble](<https://github.com/DaveGamble>)

Current Maintainer:

- [Max Bruckner](<https://github.com/FSMaxB>)

- [Alan Wang](<https://github.com/Alanscut>)

Contributors:

- \* [Ajay Bhargav](<https://github.com/ajaybhargav>)
- \* [Alexander Vasiljev](<https://github.com/AlexanderVasiljev>)
- \* [Alper Akcan](<https://github.com/alperakcan>)
- \* [Andrew Tang](<https://github.com/singku>)
- \* [Andy](<https://github.com/mlh0101>)
- \* [Anton Sergeev](<https://github.com/anton-sergeev>)
- \* [Benbuck Nason](<https://github.com/bnason-nf>)
- \* [Bernt Johan Damslara](<https://github.com/bjda>)
- \* [Bob Kocisko](<https://github.com/bobkocisko>)
- \* [Christian Schulze](<https://github.com/ChristianSch>)
- \* [Casperinous](<https://github.com/Casperinous>)
- \* [ChenYuan](<https://github.com/zjuchenyuan>)
- \* [Debora Grosse](<https://github.com/DeboraG>)
- \* [dieyushi](<https://github.com/dieyushi>)
- \* [Dngwn Hung ()](<https://github.com/DongwenHuang>)
- \* [Donough Liu](<https://github.com/ldm0>)
- \* [Erez Oxman](<https://github.com/erez-o>)
- \*

Eswar Yaganti

- \* [Evan Todd](<https://github.com/etodd>)
- \* [Fabrice Fontaine](<https://github.com/ffontaine>)
- \* Ian Mobley
- \* Irwan Djadjadi
- \* [hopper-vul](<https://github.com/hopper-vul>)
- \* [HuKeping](<https://github.com/HuKeping>)
- \* [IvanVoid](<https://github.com/npi3pak>)

- \* [Jakub Wilk](https://github.com/jwilk)
- \* [Jiri Zouhar](https://github.com/loigu)
- \* [Jonathan Fether](https://github.com/jfether)
- \* [Joshua Arulsamy](https://github.com/jarulsamy)
- \* [Julian Ste](https://github.com/julian-st)
- \* [Julin Vsquez](https://github.com/juvasquezg)
- \* [Junbo Zheng](https://github.com/Junbo-Zheng)
- \* [Kevin Branigan](https://github.com/kbranigan)
- \* [Kevin Sapper](https://github.com/sappo)
- \* [Kyle Chisholm](https://github.com/ChisholmKyle)
- \* [Linus Wallgren](https://github.com/ecksun)
- \* [Luo Jin](https://github.com/Up-wind)
- \* [Max](https://github.com/maebex)
- \* [MaxBrandtner](https://github.com/MaxBrandtner)
- \* [Mateusz Szafoni](https://github.com/raiden00pl)
- \* Mike Pontillo
- \* [miaoerduo](https://github.com/miaoerduo)
- \*
- [mohawk2](https://github.com/mohawk2)
- \* [Mike Jerris](https://github.com/mjerris)
- \* [Mike Robinson](https://github.com/mhrobinson)
- \* [Moorthy](https://github.com/moorthy-bs)
- \* [myd7349](https://github.com/myd7349)
- \* [NancyLi1013](https://github.com/NancyLi1013)
- \* [Orri](https://github.com/sbvoxel)
- \* Paulo Antonio Alvarez
- \* [Pawe Malowany](https://github.com/PawelMalowany)
- \* [Pawel Winogrodzki](https://github.com/PawelWMS)
- \* [prefetchnta](https://github.com/prefetchnta)
- \* [Rafael Leal Dias](https://github.com/rafaeldias)
- \* [Randy](https://github.com/randy408)
- \* [raiden00pl](https://github.com/raiden00pl)
- \* [Robin Mallinson](https://github.com/rmallins)
- \* [Rod Vagg](https://github.com/rvagg)
- \* [Roland Meertens](https://github.com/rmeertens)
- \* [Romain Porte](https://github.com/MicroJoe)
- \* [SANJEEV BA](https://github.com/basanjeev)
- \* [Sang-Heon Jeon](https://github.com/Intuition)
- \* [Sayan Bandyopadhyay](https://github.com/saynb)
- \* [Simon Sobisch](https://github.com/GitMensch)
- \*
- [Simon Ricaldone](https://github.com/simon-p-r)
- \* [Stoian Ivanov](https://github.com/sdrsdr)
- \* [SuperH-0630](https://github.com/SuperH-0630)
- \* [Square789](https://github.com/Square789)
- \* [Stephan Gatzka](https://github.com/gatzka)
- \* [Tony Langhammer](https://github.com/BigBrainAFK)
- \* [Vemake](https://github.com/vemakereporter)

- \* [Wei Tan](https://github.com/tan-wei)
- \* [Weston Schmidt](https://github.com/schmidtw)
- \* [xiaomianhehe](https://github.com/xiaomianhehe)
- \* [yangfl](https://github.com/yangfl)
- \* [yuta-oxo](https://github.com/yuta-oxo)
- \* [Zach Hindes](https://github.com/zhindes)
- \* [Zhao Zhixu](https://github.com/zhaozhixu)
- \* [10km](https://github.com/10km)

And probably more people on [SourceForge](https://sourceforge.net/p/cjson/bugs/search/?q=status%3Aclosed-rejected+or+status%3Aclosed-out-of-date+or+status%3Awont-fix+or+status%3Aclosed-fixed+or+status%3Aclosed&page=0)

Also thanks to all the people who reported bugs and suggested new features.

Apache License 2.0

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.123 curl 8.14.1-r1

## 1.123.1 Available under license :

### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2025, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2025 Cisco Systems, Inc. All rights reserved.