The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided “as is” with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice.

In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

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. 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. (1721R)

© 2016–2017 Cisco Systems, Inc. All rights reserved.
## Changes to This Document

This table lists the technical changes made to this document since it was first released.

*Table 1: Changes to This Document*

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2017</td>
<td>Republished for Cisco IOS XR Release 6.3.1</td>
</tr>
<tr>
<td>July 2017</td>
<td>Republished for Cisco IOS XR Release 6.2.2</td>
</tr>
<tr>
<td>November 2016</td>
<td>Republished for Cisco IOS XR Release 6.1.2.</td>
</tr>
<tr>
<td>December 2015</td>
<td>First release for Cisco IOS XR Release 6.0.0</td>
</tr>
</tbody>
</table>

Flexible Packaging Configuration Guide for Cisco ASR 9000 Series Routers
For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.

© 2016–2017 Cisco Systems, Inc. All rights reserved.
CONTENTS

Changes to This Document iii

CHAPTER 1 New and Changed Feature Information 1
    New and Changed Feature Information 1

CHAPTER 2 Introduction to Flexible Packaging 3
    Workflow for Flexible Packaging 4
    Packaging Filename Format 4

CHAPTER 3 Manage Automatic Dependency 7
    Update RPMs and SMUs 8
    Upgrade Base Software Version 8

CHAPTER 4 Customize Installation using Golden ISO 11
    Limitations 11
    Golden ISO Workflow 12
    Build Golden ISO 12
    Install Golden ISO 16
New and Changed Feature Information

This section lists all the new and changed features for the Flexible Packaging Configuration Guide.

- New and Changed Feature Information, on page 1

# New and Changed Feature Information

Table 2: New and Changed Features in Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new features were introduced.</td>
<td>NA</td>
<td>Release 6.3.1</td>
<td>NA</td>
</tr>
</tbody>
</table>
Flexible packaging is a method of breaking down the Cisco IOS XR operating system into modules and providing them as RPMs (packages). Delivering packages as RPMs enables easier and faster system updates.

The lean base software contains only the required packages, while the optional packages are provided separately as installable RPMs. You can select and install the services you want by selecting the required RPMs.

Flexible packaging feature also supports automatic dependency management, whereby, while the user is updating an RPM, the system automatically identifies all relevant dependent packages and updates them. The system uses standard LINUX tools to manage dependency during upgrades.

The base software is the minimum mandatory package (with utilities), required for the basic functioning of the router. This is also called the mini.iso file. This base package contains:

- Operating system (OS)—Kernel, file system, memory management, and other OS utilities
- Base components—Interface manager, system database, checkpoint services, configuration management utilities
- Infrastructure components—rack management, fabric management
- Routing protocols—mandatory routing protocols (such as BGP)
- Forwarding components—FIB, ARP, QoS, ACL
- Line card drivers

Mandatory RPMs (such as, BGP) which are a part of the base software, cannot be removed and can only be upgraded. Optional RPMs such as, EIGRP can be added, upgraded and removed as required.

This section also describes the following:
Workflow for Flexible Packaging

This image shows the overall workflow for Flexible Packaging.

**Figure 2: Flexible Packaging Workflow**

Packaging Filename Format

The format of an RPM is: **name-version-release.architecture.rpm** where,

- **name** - of the platform the software supports
- **version** - the version of the software
- **release** - the number of times this version of the software has been delivered
- **architecture** - the node's processor architecture

Consider the following example:

```plaintext
asr9k-mpls-1.0.2.0-r611.x86_64.rpm
```

**Platform-Package Name : asr9k-mpls**
**Version : 1.0.2.0**
**Release : r611**
**Architecture : x86_64**

Software Maintenance Upgrades (SMUs) are delivered as RPMs. RPMs have a four-digit version number. The first three digits represent major, minor, and build numbers respectively. The fourth digit is incremented with each SMU release.
This table lists the reasons when each digit of the version gets incremented.

<table>
<thead>
<tr>
<th>Version (Digit from left)</th>
<th>Indicates</th>
<th>Incremented When</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Major</td>
<td>non-backward compatible API(s) change(s)</td>
</tr>
<tr>
<td>Second</td>
<td>Minor</td>
<td>a backward compatible change occurs to a public API</td>
</tr>
<tr>
<td>Third</td>
<td>Build</td>
<td>an RPM is built without any API change</td>
</tr>
<tr>
<td>Fourth</td>
<td>SMU Release</td>
<td>a new SMU is released</td>
</tr>
</tbody>
</table>

**Defect ID**

SMUs are identified with a defect-ID. In this example, note that, for the first SMU release of the package, the fourth digit starts at 1 and for the second SMU release of the package, the fourth digit is incremented to 2.

First SMU of the mpls package: asr9k-mpls-1.0.2.1-r611.CSCub12345.x86_64.rpm
Second SMU of the mpls package: asr9k-mpls-1.0.2.2-r611.CSCub12345.x86_64.rpm
CHAPTER 3

Manage Automatic Dependency

Flexible packaging supports automatic dependency management. While the user is updating an RPM, the system automatically identifies all relevant dependent packages and updates them.

*Figure 3: Flow for Installation (base software, RPMs and SMUs)*

Until this release, users downloaded the software image and required RPMs from CCO on a network server (the repository). They used the `install add` and the `install activate` commands to add and activate the downloaded files on the . Then, users needed to manually identify relevant dependent RPMs, to add and activate them.

With automatic dependency management, users need not identify dependent RPMs to individually add and activate them. They can execute new install commands to identify and install dependent RPMs automatically.

The new commands are `install update` and `install upgrade`. The `install update` command identifies and updates dependent packages. The command does not update the base package. The `install upgrade` command upgrades the base package.

*Note:

1. Cisco IOS XR Version 6.1.1 does not provide 3rd-party SMUs as part of automatic dependency management (`install update` and `install upgrade` commands). The 3rd party SMUs must be installed separately, and in isolation from other installation procedures (installation of SMUs and RPMs in IOS XR or admin containers).

2. Cisco IOS XR Version 6.1.1 does not support asynchronous package upgrades.*

The rest of this chapter contains these sections:

- Update RPMs and SMUs, on page 8
An RPM may contain a fix for a specific defect, and you may need to update the system with that fix. To update RPMs and SMUs to a newer version, use the **install update** command. When the **install update** command is issued for a particular RPM, the router communicates with the repository, and downloads and activates that RPM. If the repository contains a dependent RPM, the router identifies that dependent RPM and installs that too.

The syntax of the **install update** command is:

```
install update source  repository [rpm]
```

Four scenarios in which you can use the **install update** command are:

- **When a package name is not specified**

  When no package is specified, the command updates the latest SMUs of all installed packages.

  ```
  install update source [repository]
  ```

- **When a package name is specified**

  If the package name is specified, the command installs that package, updates the latest SMUs of that package, along with its dependencies. If the package is already installed, only the SMUs of that package are installed. (SMUs that are already installed are skipped.)

  ```
  install update source [repository] asr9k-mpls.rpm
  ```

- **When a package name and version number are specified**

  If a particular version of package needs to be installed, the complete package name must be specified; that package is installed along with the latest SMUs of that package present in the repository.

  ```
  install update source [repository] asr9k-mpls-1.0.2.0-r611.x86_64.rpm
  ```

- **When an SMU is specified**

  If an SMU is specified, that SMU is downloaded and installed, along with its dependent SMUs.

  ```
  install update source [repository] asr9k-mpls-1.0.2.1-r611.CSCub12345.x86_64.rpm
  ```

---

**Upgrade Base Software Version**

You may choose to upgrade to a newer version of the base software when it becomes available. To upgrade to the latest base software version, use the **install upgrade** command. With the upgrade of the base version, RPMs that are currently available on the router are also upgraded.

---

**Note**

SMUs are not upgraded as part of this process.

The syntax of the **install upgrade** command is:

```
install upgrade source  repository version version [rpm]
```
VRF and TPA on dataport is not supported. If the server is reachable only through non-default VRF interface, the file must already be retrieved using ftp, sftp, scp, http or https protocols.

You can use the `install upgrade` command when:

- **The version number is specified**
  
The base software (.mini) is upgraded to the specified version; all installed RPMs are upgraded to the same release version.

  `install upgrade source [repository] version 6.2.2`
CHAPTER 4

Customize Installation using Golden ISO

Golden ISO (GISO) is a customized ISO that a user can build to suit the installation requirement. The user can customize the installable image to include the standard base image with the basic functional components, and add additional RPMs, SMUs and configuration files based on requirement.

The ease of installation and the time taken to seamlessly install or upgrade a system plays a vital role in a cloud-scale network. An installation process that is time-consuming and complex affects the resiliency and scale of the network. The GISO simplifies the installation process, automates the installation workflow, and manages the dependencies in RPMs and SMUs automatically.

GISO is built using a build script `gisobuild.py` available on the github location https://github.com/ios-xr/gisobuild. For more information about the build script and the steps to build GISO, see Build Golden ISO, on page 12.

When a system boots with GISO, additional SMUs and RPMs in GISO are installed automatically, and the router is pre-configured with the XR configuration in GISO. For more information about downloading and installing GISO, see Install Golden ISO, on page 16.

The capabilities of GISO can be used in the following scenarios:

- Migration from IOS XR 32-bit to IOS XR 64-bit
- Initial deployment of the router
- Software disaster recovery
- System upgrade from one base version to another
- System upgrade from same base version but with additional SMUs
- Install update to identify and update dependant packages

Limitations

The following are the known problems and limitations with the customized ISO:
• Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.
• GISO build script `gisobuild.py` does not support verifying the XR configuration.
• Renaming a GISO build and then installing from the renamed GISO build is not supported.

**Golden ISO Workflow**

The following image shows the workflow for building and installing golden ISO.

---

**Build Golden ISO**


The GISO build script supports automatic dependency management, and provides these functionalities:

• Builds RPM database of all the packages present in package repository.
• Skips and removes Cisco RPMs that do not match the mini-x.iso version.

• Skips and removes third-party RPMs that are not SMUs of already existing third-party base package in mini-x.iso.

• Displays an error and exits build process if there are multiple base RPMs of same release but different versions.

• Performs compatibility check and dependency check for all the RPMs. For example, the child RPM asr9000-mpls-te-rsvp is dependent on the parent RPM asr9000-mpls. If only the child RPM is included, the Golden ISO build fails.

To build GISO, provide the following input parameters to the script:

• Base mini-x.iso (mandatory)

• XR configuration file (optional)

• one or more Cisco-specific SMUs for host, XR and System admin

• one or more third-party SMUs for host, XR and System admin

• Label for golden ISO

---

Note

Golden ISO can be built only from mini ISO. The full or fullk9 bundle ISO is not supported.

Use the following naming convention when building GISO:

<table>
<thead>
<tr>
<th>GISO Build</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>GISO without k9sec RPM</td>
<td>&lt;platform-name&gt;-golden-x.iso-&lt;version&gt;.&lt;label&gt;</td>
<td>&lt;platform-name&gt;-golden-x.iso-&lt;version&gt;.v1</td>
</tr>
<tr>
<td></td>
<td>&lt;platform-name&gt;-golden-x-&lt;version&gt;.iso.&lt;label&gt;</td>
<td>&lt;platform-name&gt;-golden-x-&lt;version&gt;.iso.v1</td>
</tr>
<tr>
<td>GISO with k9sec RPM</td>
<td>&lt;platform-name&gt;-goldenk9-x.iso-&lt;version&gt;.&lt;label&gt;</td>
<td>&lt;platform-name&gt;-goldenk9-x.iso-&lt;version&gt;.v1</td>
</tr>
<tr>
<td></td>
<td>&lt;platform-name&gt;-goldenk9-x-&lt;version&gt;.iso.&lt;label&gt;</td>
<td>&lt;platform-name&gt;-goldenk9-x-&lt;version&gt;.iso.v1</td>
</tr>
</tbody>
</table>

Note

To successfully add k9sec RPM to GISO, change the permission of the file to 644 using the `chmod` command.

```bash
chmod 644 [k9 sec rpm]
```

To build GISO, perform the following steps:

**Before you begin**

• To upgrade from non-GISO to GISO version, it is mandatory to first upgrade to mini ISO with GISO support. For ASR 9000 series routers, upgrade to release 6.1.3 or later.

• The system where GISO is built must meet the following requirements:
  • System must have Python version 2.7 and later.
• System must have free disk space of minimum 3 to 4 GB.
• Verify that the Linux utilities `mount`, `rm`, `cp`, `umount`, `zcat`, `chroot`, `mkisofs` are present in the system. These utilities will be used by the script. Ensure privileges are available to execute all of these Linux commands.
• Kernel version of the system must be later than 3.16 or later than the version of kernel of Cisco ISO.
• Verify that a `libyaml` rpm supported by the Linux kernel is available to successfully `import yaml` in the tool.
• User should have proper permission for security rpm(`k9sec-rpm`) in rpm repository, else security rpm would be ignored for Golden ISO creation.

• The system from where the gisobuild script is executed must have root credentials.

---

**Step 1**  
Copy the script `gisobuild.py` from the github location [https://github.com/ios-xr/gisobuild](https://github.com/ios-xr/gisobuild) to an offline system or external server where the GISO will be built. Ensure that this system meets the pre-requisites described above in the *Before You Begin* section.

**Step 2**  
Run the script `gisobuild.py` and provide parameters to build the golden ISO off the router. Ensure that all RPMs and SMUs are present in the same directory. The number of RPMs and SMUs that can be used to build the Golden ISO is 128.

**Note**  
The `-i` option is mandatory, and either or both `-r` or `-c` options must be provided.

```
```

Golden ISO build process starting...

System requirements check [PASS]

Platform: asr9k Version: 6.1.3

Info: Migration option is provided so migration tar will be generated

Scanning repository [repo]...

Building RPM Database...

Total 56 RPM(s) present in the repository path provided in CLI

Following XR x86_64 rpm(s) will be used for building Golden ISO:

```plaintext
(+) asr9k-diags-x64-1.0.0.0-r613.x86_64.rpm
(+) asr9k-parser-x64-2.0.0.0-r613.x86_64.rpm
(+) asr9k-mgb1-x64-3.0.0.0-r613.x86_64.rpm
(+) asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm
(+) asr9k-os-supp=64-4.0.0.1-r613.CSChu77777.x86_64.rpm
(+) asr9k-mls-x64-2.1.0.0-r613.x86_64.rpm
(+) asr9k-k9sec-x64-2.2.0.1-r613.CSCxr33333.x86_64.rpm
```

---------- Truncated

Skipping following rpms from repository since they are already present in base ISO:

```plaintext
(-) asr9k-parser-x64-2.0.0.0-r613.x86_64.rpm
(-) asr9k-bgp-x64-1.0.0.0-r613.x86_64.rpm
(-) asr9k-diags-x64-1.0.0.0-r613.x86_64.rpm
```
RPM compatibility check [PASS]

Following SYSADMIN x86_64 rpm(s) will be used for building Golden ISO:

(+) asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
(+) asr9k-sysadmin-shared-6.1.3-r613.CSCcv22222.x86_64.rpm
(+) asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64.rpm

------------Truncated

RPM compatibility check [PASS]

Following HOST x86_64 rpm(s) will be used for building Golden ISO:

(+) asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64.rpm
(+) cisco-klm-mifpga-0.1.p1-r0.0.CSCtp12345.host.x86_64.rpm
(+) kernel-modules-3.14-r0.1.host.x86_64.rpm

...RPM compatibility check [PASS]

Building Golden ISO...

Summary ......

XR rpms:
  asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm
  iosxr-infra-asr9k-64-4.0.0.2-r613.CSCxr11111.x86_64.rpm
  asr9k-mpls-te-rsvp-x64-1.2.0.0-r613.x86_64.rpm

HOST rpms:
  asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64.rpm
  cisco-klm-mifpga-0.1.p1-r0.0.CSCtp12345.host.x86_64.rpm
  kernel-modules-3.14-r0.1.host.x86_64.rpm

...Golden ISO creation SUCCESS.

Golden ISO Image Location: <repo>/asr9k-goldenk9-x64.iso-6.1.3.v1

Building Migration tar...

Migration tar creation SUCCESS.

Migration tar Location: <repo>/asr9k-goldenk9-x64-migrate_to_eXR.tar-6.1.3.v1

Detail logs: <repo>/Giso_build.log-2016-12-18:03:54:53.837680
[03:54:53]—[router:<repo>]$ ls -ltr asr9k-goldenk9-x64*
-rw-r--r-- 1 root root 1274669056 Dec 18 03:54 asr9k-goldenk9-x64.iso-6.1.3.v1
-rw-r--r-- 1 root root 1425008640 Dec 18 03:54 asr9k-goldenk9-x64-migrate_to_eXR.tar-6.1.3.v1

----------Truncated

where:

- -i is the path to mini-x.iso
- -r is the path to RPM repository
- -c is the path to XR config file
- -l is the golden ISO label
- -h shows the help message
- -v is the version of the build tool gisobuild.py
• -m is to build the migration tar to migrate from IOS XR to IOS XR 64 bit

GISO is built with the RPMs placed in respective folders in the specified directory and also includes the log files giso_summary.txt and gisobuild.log-<timestamp>. The XR configuration file is placed as router.cfg in the directory.

Note
The GISO script does not support verification of XR configuration.

What to do next
Install the golden ISO on the router.

Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

• Installs host and system admin RPMs.
• Partitions repository and TFTP boot on RP.
• Creates software profile in system admin and XR modes.
• Installs XR RPMs. Use show instal active command to see the list of RPMs.
• Applies XR configuration. Use show running-config command in XR mode to verify.

Step 1
Download GISO image to the router using one of the following options:

• PXE boot: when the router is booted, the boot mode is identified. After detecting PXE as boot mode, all available ethernet interfaces are brought up, and DHClient is run on each interface. DHClient script parses HTTP or TFTP protocol, and GISO is downloaded to the box.

The following example shows the logs from installation of GISO using PXE boot:

... Fri Dec 2 19:18:03 UTC 2016: ---Starting to prepare host logical volume---
... Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.host.x86_64.rpm) from installation
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.host.x86_64.rpm) from installation
Fri Dec 02 19:18:15 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64 [SUCCESS]
...

Fri Dec 2 19:18:23 UTC 2016: ---Starting to prepare calvados logical volume---
...

Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.admin.x86_64.rpm) from installation
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.admin.x86_64.rpm)
Customize Installation using Golden ISO

Install Golden ISO

from installation
Fri Dec 02 19:18:49 UTC 2016: Installing asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64 [SUCCESS]
Fri Dec 02 19:18:50 UTC 2016: Installing asr9k-sysadmin-shared-6.1.3-r613.CSCcv33333.x86_64 [SUCCESS]
Fri Dec 02 19:18:51 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.admin.x86_64 [SUCCESS]

... 
Fri Dec 2 19:19:07 UTC 2016: ---Starting to prepare repository---
Fri Dec 2 19:19:11 UTC 2016: File system creation on /test took 3 seconds
Fri Dec 2 19:19:11 UTC 2016: Copying /iso/host.iso to repository /iso directory
Fri Dec 2 19:19:13 UTC 2016: Copying /iso/asr9k-sysadmin.iso to repository /iso directory
Fri Dec 2 19:19:13 UTC 2016: Copy Sysadmin rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy Host os rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy XR rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy giso_info.txt to repository
Fri Dec 2 19:19:17 UTC 2016: Copying /iso/asr9k-xr.iso to repository /iso directory
Fri Dec 2 19:19:21 UTC 2016: Copying all ISOs to repository took 10 seconds

... 

• **USB boot or Disk Boot:** when the USB mode is detected during boot, and GISO is identified, the additional RPMs and XR configuration files are extracted and installed.

• **System Upgrade** when the system is upgraded, GISO can be installed using `install add`, `install activate`, or using `install update` commands.

  • **system upgrade from a non-GISO (image that does not support GISO) to GISO image:** If a system is running a version 1 with an image that does not support GISO, the system cannot be upgraded directly to version 2 of an image that supports GISO. Instead, the version 1 must be upgraded to version 2 mini ISO, and then to version 2 GISO.

  • **system upgrade in a release from version 1 GISO to version 2 GISO:** If both the GISO images have the same base version but different labels, `install add` and `install activate` commands does not support same version of two images. Instead, using `install update` command installs only the delta RPMs. System reload is based on restart type of the delta RPMs.

Router#install update source <path-to-image> <platform-name-goldenk9-x64.iso-6.1.3>
Sat Dec  3 15:51:43.384 UTC

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ISO <platform-name-goldenk9-x64.iso-6.1.3> in input package list. Going to upgrade the system to version 6.1.3.2.

Updating contents of golden ISO

Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : Dir
Collecting software state..

Fetching .... <asr9k-name-goldenk9-x64.iso-6.1.3.v2>
LC/0/0/CPU0: Dec 3 15:53:41.580 : %PLATFORM-STATS-INFRA-3-ERR_STR_1 : ErrStr: Unable to map stats infra shared mem

Skipping openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm from GISO as it's active
Skipping <platform-name>-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm from GISO as it's active
Skipping openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm from GISO as it's active
Skipping openssh-scp-6.6p1-r0.0.host.x86_64.rpm from GISO (skipped SMUs base pkg)

Adding packages

<platform-name>-sysadmin-shared-6.1.3-r613.CSCCv33333.x86_64.rpm
<platform-name>-mpls-te-rsvp-x64-1.2.0.0-r613.x86_64.rpm

Router: Dec 3 15:54:12.308 : sdr_instmgr[1145]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 1 finished successfully

Dec 03 15:54:03 Install operation 1 started by root:

install add source <repository-location>/6.1.3
asr9k-sysadmin-shared-6.1.3-r613.CSCCv33333.x86_64.rpm
asr9k-mpls-te-rsvp-x64-1.2.0.0-r613.x86_64.rpm

Dec 03 15:54:05 Install operation will continue in the background

Dec 03 15:54:13 Install operation 1 finished successfully

--------Truncated

- **system upgrade across releases from version1 GISO to version2 GISO:** Both the GISO images have different base versions. Use `install add` and `install activate` commands, or `install update` command to perform the system upgrade. The router reloads after the upgrade with the version2 GISO image.

Router# install update source <path-to-image> <platform-name-goldenk9-x64.iso-6.1.3.opt

Sat Dec 3 08:11:46.198 UTC
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ISO <platform-name-goldenk9-x64.iso-6.1.3.opt> in input package list. Going to upgrade the system to version 6.1.3.
Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : Dir
Collecting software state..
Getting platform
Getting supported architecture
Getting active packages from XR
Getting inactive packages from XR
Getting list of RPMs in local repo
Getting list of provides of all active packages
Getting requires of each rpm in repo
LC/0/0/CPU0: Dec 3 08:16:25.147 : fab_si[193]: %PLATFORM-STATS-INFRA-3-ERR_STR_1 : ErrStr: Unable to map stats infra shared mem
Fetching .... <platform-name-goldenk9-x64.iso-6.1.3.opt>
Adding packages
<platform-name>-goldenk9-x64.iso-6.1.3.opt

.......
Dec 3 08:17:07 Install operation 2 started by root:
Install add source <path>/6.1.3 <platform-name>-goldenk9-x64.iso-6.1.3.opt

Dec 3 08:17:08 Install operation will continue in the background

Dec 3 08:26:43 Install operation 2 finished successfully

Install add operation successful
Activating <platform-name>-goldenk9-x64-6.1.3

Dec 3 08:26:49 Install operation 3 started by root:
   install activate pkg <platform-name>-goldenk9-x64-6.1.3

Dec 3 08:26:49 Package list:
Dec 3 08:26:49 <platform-name>-goldenk9-x64-6.1.3
This install operation will reload the sdr, continue?
[yes/no]:[yes] Dec 3 08:27:10 Install operation will continue in the background

------Truncated

Step 2 Run the show install repository all command in System Admin mode to view the RPMs and base ISO for host, system admin and XR.

sysadmin-vm:0_RP0# show install repository all
Admin repository
---------------------
asr9k-sysadmin-6.1.1
asr9k-sysadmin-hostos-6.1.1-r611.CSCcv10001.admin.x86_64
asr9k-sysadmin-system-6.1.1-r611.CSCcv10005.x86_64
....
XR repository
------------------
asr9k-iosxr-mgbl-3.0.0.0-r611.x86_64
asr9k-xr-6.1.1
....
Host repository
---------------------
host-6.1.1

Step 3 Run the show install package <golden-iso> command to display the list of RPMs, and packages built in GISO.

Router#show install package asr9k-goldenk9-x64-6.1.3

Sun Dec  4 13:52:48.279 UTC
This may take a while ...
ISO Name: asr9k-goldenk9-x64-6.1.3
ISO Type: bundle
ISO Bundled: asr9k-mini-x64-6.1.3
Golden ISO Label: temp
ISO Contents:
ISO Name: asr9k-xr-6.1.3
ISO Type: x
rpms in x ISO:
   iosxr-os-asr9k-64-5.0.0.0-r613
   iosxr-ce-asr9k-64-3.0.0.0-r613
   iosxr-infra-asr9k-64-4.0.0.0-r613
   iosxr-fwding-asr9k-64-4.0.0.0-r613
   iosxr-routing-asr9k-64-3.1.0.0-r613
   ...
ISO Name: asr9k-sysadmin-6.1.3
ISO Type: sysadmin
rpms in sysadmin ISO:
asr9k-sysadmin-topo-6.1.3-r613
asr9k-sysadmin-shared-6.1.3-r613
asr9k-sysadmin-system-6.1.3-r613
asr9k-sysadmin-hostos-6.1.3-r613.admin
...

ISO Name: host-6.1.3
ISO Type: host
rpms in host ISO:
  asr9k-sysadmin-hostos-6.1.3-r613.host

Golden ISO Rpms:
xr rpms in golden ISO:
  asr9k-k9sec-x64-2.2.0.1-r613.CSCxr33333.x86_64.rpm
  openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
  openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
  asr9k-mpls-x64-2.1.0.0-r613.x86_64.rpm
  asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm

sysadmin rpms in golden ISO:
  asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
  openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
  openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm

host rpms in golden ISO:
  openssh-scp-6.6p1-r0.0.host.x86_64.rpm
  openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm

The ISO, SMUs and packages in GISO are installed on the router.