



Cisco IOS XR Flexible Packaging Configuration Guide for Cisco NCS 1000 Series

First Published: 2016-05-05

Last Modified: --

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number:

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

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.

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: <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)

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.

© Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Flexible Packaging - Easy Routine Upgrades and Maintenance 1

Workflow for Flexible Packaging 3

Packaging Filename Format 3

CHAPTER 2

Manage Automatic Dependency 5

Manage Automatic Dependency 6

Update RPMs and SMUs 7

Upgrade Base Software Version 7

Downgrade an RPM 8



CHAPTER

1

Flexible Packaging - Easy Routine Upgrades and Maintenance

Flexible packaging is an enhancement that modularizes the Cisco IOS XR operating system as RPM packages. Redhat Packet Manager (RPM) based delivery of packages enable easier and faster system updates.

The base software is leaner, containing only required mandatory packages. Other optional packages are separated and made available as individually installable RPM packages. Users have the flexibility to select and install the services they want, by choosing relevant optional RPMs.

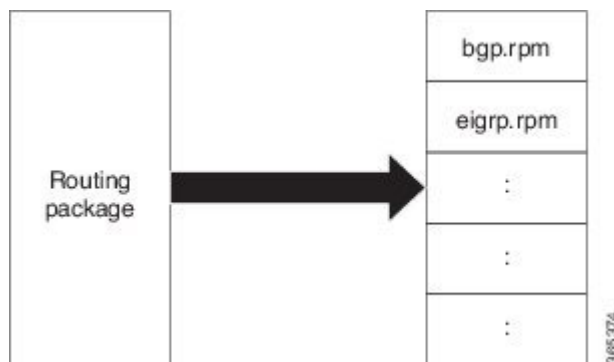
Flexible packaging also supports automatic dependency management, where, while the user is updating an RPM, the system identifies all relevant dependent packages and updates them. The system uses standard LINUX tools to manage dependency during upgrades.

The base software is the minimum mandatory package (with utilities), required for the basic functioning of the NCS 1002. This is also called the mini.iso. 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, ISIS)
- 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.

Figure 1: Granular routing modules



The RPMs that are currently available in the granular module format are - mpls-te-rsvp, bgp (upgrade only),

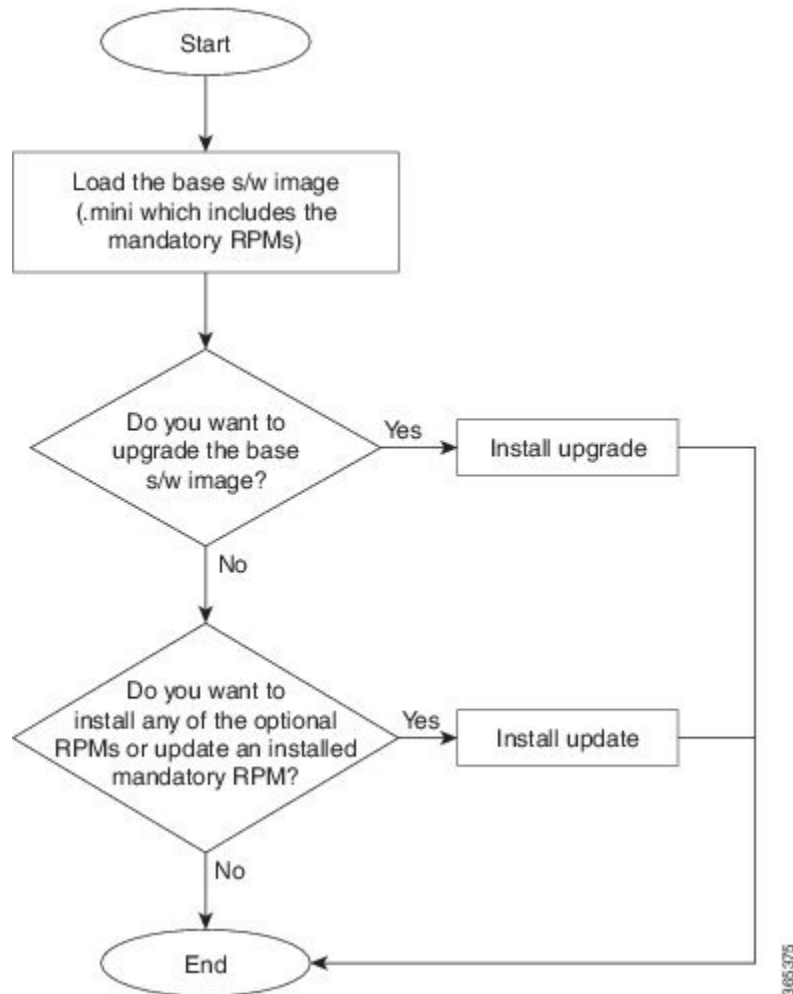
This chapter also discusses:

- [Workflow for Flexible Packaging, page 3](#)
- [Packaging Filename Format, page 3](#)

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 this example: -mpls-1.2.0.0-r622.x86_64.rpm

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.

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

Defect ID

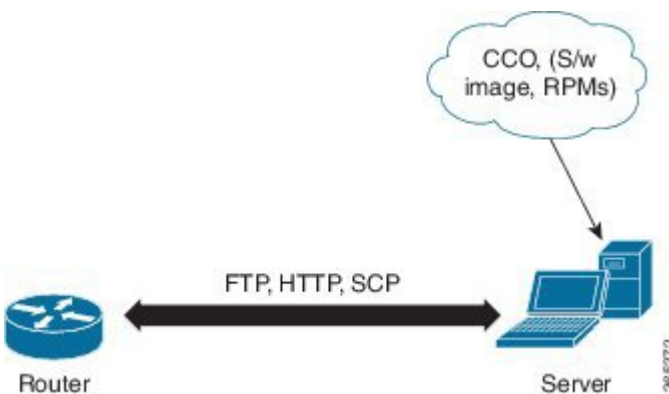
SMUs are identified with a defect-ID. In this example, note that, for the first SMU release of the package, the fourth digit starts at 1 and for the second SMU release of the package, the fourth digit is incremented to 2.



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 NCS 1002. 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 6.0.1 does not provide 3rd-party and host package SMUs as part of automatic dependency management (**install update** and **install upgrade** commands). The 3rd party and host package 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 6.0.1 does not yet support asynchronous package upgrades.

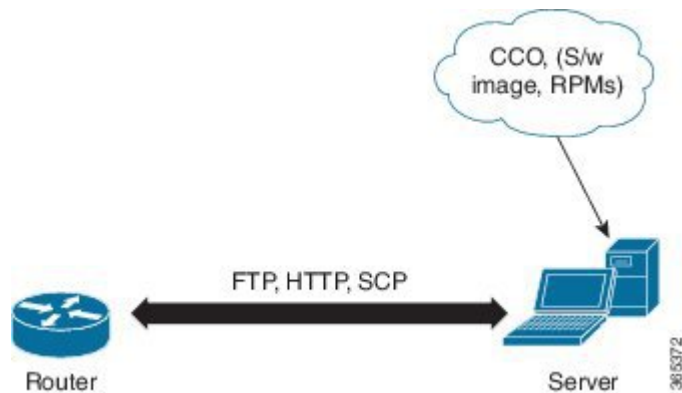
The rest of this chapter contains these sections:

- [Manage Automatic Dependency, page 6](#)
- [Update RPMs and SMUs, page 7](#)
- [Upgrade Base Software Version, page 7](#)
- [Downgrade an RPM, page 8](#)

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 4: 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 NCS 1002. 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 6.0.1 does not provide 3rd-party and host package SMUs as part of automatic dependency management (**install update** and **install upgrade** commands). The 3rd party and host package 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 6.0.1 does not yet support asynchronous package upgrades.

The rest of this chapter contains these sections:

Update RPMs and SMUs

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 NCS 1002 communicates with the repository, and downloads and activates that RPM. If the repository contains a dependent RPM, the NCS 1002 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] ncs1002-mp1s.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] ncs1002-mp1s-1.0.0.1-r622.CSCub12345.x86_64.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] ncs1002-mp1s-1.0.2.0-r622.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 NCS 1002 are upgraded too.

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

install upgrade source *repository version version [rpm]*

Two scenarios in which you can use the **install upgrade** command are:

- **When 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
```

- **When a version number for an RPM is specified**

When performing a system upgrade, the user can choose to have an optional RPM to be of a different release (from that of the base software version); that RPM can be specified.

```
install upgrade source [repository] version 6.2.2 ncs1002-mpls-1.0.2.0-r623.x86_64.rpm
```

Downgrade an RPM

After an RPM is activated, you may need to downgrade it by activating an RPM of a lower version. Use the force option with the **install activate** command to activate an RPM of a lower version.

The syntax of the command is: **install activate [rpm] force**

Example

Purpose: To add and activate an RPM of a lower version

RPM currently active: mpls-te-rsvp-2.0.0.0-r60011I

RPM to be activated: mpls-te-rsvp-2.0.0.0-r6006I

Configuration

- 1 Download the RPM, **mpls-te-rsvp-2.0.0.0-r6006I.rpm**, to the NCS 1002.

```
install add source [repository] mpls-te-rsvp-2.0.0.0-r6006I.rpm
```
- 2 Activate the RPM, **mpls-te-rsvp-2.0.0.0-r6006I.rpm**.

```
install activate mpls-te-rsvp-2.0.0.0-r6006I.rpm force
```

After the upgrade, **mpls-te-rsvp-2.0.0.0-r6006I.rpm** is active and **mpls-te-rsvp-2.0.0.0-r60011I.rpm** is automatically rendered inactive.

Use the **show install active** command to check which version of the RPM is active.