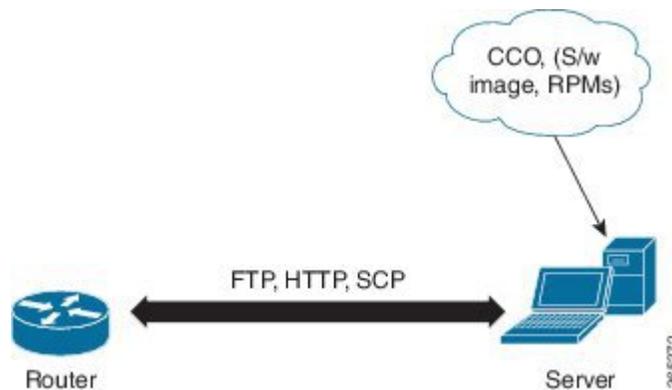




Manage Automatic Dependency

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 1: 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.

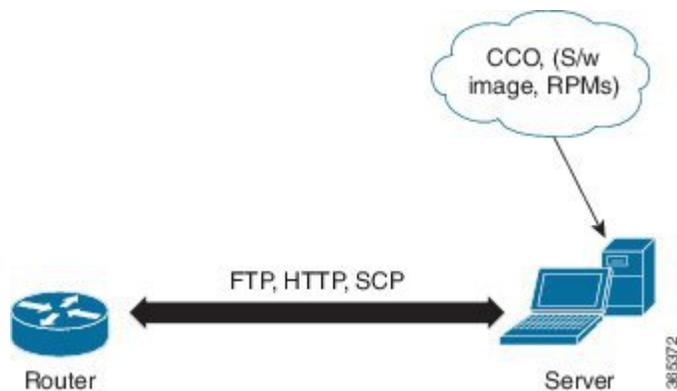
The rest of this chapter contains these sections:

- [Manage Automatic Dependency, page 2](#)
- [Update RPMs and SMUs, page 3](#)
- [Upgrade Base Software Version, page 3](#)
- [Downgrade an RPM, page 4](#)

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 2: 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-mps.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-mps-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-mps-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.