Software Management Operations

This chapter provides information about software management operations on the system.

Understanding the Local File System

The local file system on the VPC VM is made up of files that are stored on the following:

- **/flash** Flash memory allocated as vHDD-1 on the M via the hypervisor is the default storage media for the StarOS image, CLI configuration, and crash log files used by the system.
- **/hd-raid** This is the storage space allocated as vHDD-2 on the CF VM by the hypervisor. It is used to store CDRs (Charging Data Records) and UDRs (Usage Data Records).

File Types Used by the Local File System

The following file types can be located in the local file system:

- **Operating System Software Image File**: This binary file type is identified by its **.bin** extension. The file is the operating system that is loaded by the system upon startup or reloading. This is an executable, read-only file that cannot be modified by end users.
- **CLI Configuration File**: This file type is identified by its **.cfg** extension. These are text files that contain CLI commands that work in conjunction with the operating system software image. These files determine services to be provided, hardware and software configurations, and other functions performed by the system. The files are typically created by the end user. You can modify the files both on and off-line and use descriptive long filenames.

- **System File**: Only one file identified by a **.sys** extension is used by the system. The boot.sys file contains system-specific information, which describes how the system locates, and in what priority it loads, file groups (paired .bin and .cfg files) from its boot stack.
- **Abridged Crash Log**: The abridged crash log, identified by its **crashlog** filename, contains summary information about software or hardware failures that occur on the system. This file is located in the **/flash/crsh2/** directory on the device. You can view the contents of this file through the CLI, but you cannot modify the file.

Understanding the boot.sys File

The system uses the boot.sys file to store the prioritized boot stack parameters and file groups the system uses during startup. Modify this file only through system CLI commands and not through external means. Boot parameters contain information the system needs to locate the operating system image file, including:

- **bootmode**: This setting is typically configured to normal, and identifies how the system starts.
- **boot stack information**: The boot stack is made up of prioritized file group entries that designate the operating system image file and the CLI configuration file to load.

When a system is started for the first time, the boot.sys file is configured to use the normal boot mode and load the operating system software image from the /flash directory.

There is no CLI configuration file contained on the local file system. This causes the system to automatically start its CLI-based Quick Setup Wizard upon the first successful boot. Refer to *Getting Started* for more information on using the Quick Setup Wizard.

Maintaining the Local File System

Use CLI commands to manage and maintain the devices that make up the local file system. Execute all the commands described in this section in the Exec Mode. Unless otherwise specified, you must have security administrator or administrator privileges to execute these commands.

File System Management Commands

Use the commands in this section to manage and organize the local file system.

Important For complete information on the commands listed below, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference.*

Creating Directories

Use the **mkdir** command to create a new directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

`[local]`*host_name*`#` **mkdir { /flash | /usb1 | /hd-raid }** */dir_name*

Use the following command to create a directory named *configs*:

`[local]`*host_name*`#` **mkdir /flash/configs**

Renaming Files and Directories

Use the **rename** command to change the name of a file from its original name to a different name. Remember to use the same file extension, if applicable, to ensure that the file type remains unchanged.

Use the following command to rename a file named *iot_test.cfg* to *iot_accept.cfg* on the */flash* local device.

`[local]`*host_name*`#` **rename /flash/iot_test.cfg /flash/iot_accept.cfg -noconfirm**

Important Use the **rename** command only within the same local device. You cannot rename a file and place it onto another local device at the same time. To move a renamed file, you must use the **copy** command.

Copying Files

These instructions assume that you are at the root prompt for the Exec mode. To save your current configuration, enter the following command:

`[local]`*host_name*`#` **copy** *from_url to_url* **[-noconfirm]**

To copy a configuration file called *system.cfg* from a directory that was called *cfgfiles* to a directory named *configs_old*, enter the following command:

`[local]`*host_name*`#` **copy /flash/cfgfiles/system.cfg /flash/configs_old/system_2011.cfg**

To copy a configuration file called *init_config.cfg* to the root directory of a TFTP server with a hostname of *config_server*, enter the following command:

`[local]`*host_name*`#` **copy /flash/cfgfiles/init_confg.cfg tftp://config_server/init_config.cfg**

Deleting Files

The **delete** command removes a designated file from its specified location on the local file system.

Important This command does not support wildcard entries; each filename must be specified in its entirety.

Caution Do not delete the boot.sys file. If deleted, the system will not reboot on command and will be rendered inoperable.

`[local]`*host_name*`#` **delete { /flash | /usb1 | /hd-raid }**/*filename* **[-noconfirm]**

The following command deletes a file named *test.cfg* from the */flash* directory.

`[local]`*host_name*`#` **delete /flash/test.cfg**

Removing Directories

The **rmdir** command deletes a current directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

Important The directory you want to remove (delete) must be empty before executing the **rmdir** command. If the directory is not empty, the CLI displays a "Directory not empty" message and will not execute.

`[local]`*host_name*`#` **rmdir** *url /dir_name*

The following command deletes an empty directory named *configs* in the */flash* directory.

`[local]`*host_name*`#` **rmdir /flash/configs**

Formatting Local Devices

The **format** command performs a low-level format of a local device. This operation formats the device to use the FAT16 formatting method, which is required for proper read/write functionality with the operating system.

Important Local devices that have been formatted using other methods such as NTFS or FAT32 may be used to store various operating system, CLI configuration, and crash log files. However, when placing a new local device into the MIO/UMIO/MIO2 for regular use, you should format the device via the system prior to use. This ensures that the proper file allocation table format is used, preventing any possible discrepancies between other formats used with other operating systems.

Caution The **filesystem format** command removes all files and information stored on the device.

To format a local device for use by the local file system, enter the following command:

`[local]`*host_name*`#` **filesystem format { /flash | /usb1 | /hd-raid }**

Applying Pre-existing CLI Configuration Files

A pre-existing CLI configuration file is any .cfg file created to provide utility functions (such as clearing all statistics during testing) or created off-line using a text editor. There may be pre-existing configuration files stored on the local file system that can be applied to a running system at any time.

Caution If a configuration file is applied to a system currently running another CLI configuration, any like contexts, services, logical interfaces, physical ports, IP address pools, or other configured items will be overwritten if the same command exists in the configuration file being applied. Take caution to ensure that you are knowledgeable of the contents of the file being applied and understand what the service ramifications are if a currently running command is overwritten. Also note that changes will not be saved automatically.

A CLI configuration file, or script containing CLI commands, can be applied to a running system by entering the following command at the Exec mode prompt:

`[local]`*host_name*`#` **configure** *url* **[verbose]**

url specifies the location of the CLI configuration file to be applied. It may refer to a local or a remote file.

The following command applies a pre-existing CLI configuration file named *clearcmds.cfg* in the */flash* directory.

`[local]`*host_name*`#` **configure /flash/clearcmds.cfg**

Viewing Files on the Local File System

This section describes how to view a variety of files.

Viewing the Contents of a Local Device

The contents, usage information, and file system directory structure of any local device can be viewed by entering the following command at the Exec mode prompt:

directory { /flash | /usb1 | /hd-raid }

Viewing CLI Configuration and boot.sys Files

The contents of CLI configuration and boot.sys files, contained on the local file system, can be viewed off-line (without loading them into the OS) by entering the following command at the Exec mode prompt:

`[local]`*host_name*`#` **show file** *url* **{ /flash | /usb1 | /hd-raid }** *filename*

Where: *url* is the path name for the location of the file and *filename* is the name of the file, including any extension.

Important Operator and inspector-level users can execute the **show file** command but cannot execute the **directory** command.

Validating an Operating System File

The operating system software image file, identified by its .bin extension, is a non-readable, non-editable file that executes on the system, creating its runtime operating system (OS).

It is important to verify a new operating system image file before attempting to load it. To accomplish this, a proprietary checksum algorithm is used to create checksum values for each portion of the application stored within the .bin file during program compilation.

This information can be used to validate the actual file against the checksum values stored within the file during its compilation. If any portion of the image file has become corrupted (for example, the file was truncated or was transferred using ASCII mode instead of binary mode), then this information is reported and the file is deemed unusable.

To validate an operating system software image file, enter the following command at the Exec mode prompt:

`[local]`*host_name*`#` **show version { /flash | /usb1 | /hd-raid }** */[directory]/filename* **[all]**

The output of this command displays the following information:

- Version number
- Description
- Date
- Boot Image
- Size
- Flags
- Platform

If an invalid file is found, the system displays a failure message similar to these:

```
Failure: Image /flash/image_version.bin CRC check failed!
Failure: /flash/image_version.bin, has a bad magic number
```

Configuring the Boot Stack

The boot stack consists of a prioritized listing of operating system software image-to-CLI configuration file associations. These associations determine the software image and configuration file that gets loaded during system startup or upon a reload/reboot. Though multiple associations can be configured, the system uses the association with the highest priority. In the event that there is an error processing this association (for example, one of the files cannot be located), the system attempts to use the association with the next highest priority. Priorities range from 1 to 100, with 1 being the highest priority. The maximum number of boot stack entries that may be configured in the boot.sys file is 10.

Boot stack information is contained in the boot.sys file, described in Understanding the boot.sys File, on page 2. In addition to boot stack entries, the boot.sys file contains any configuration commands required to define the system boot method as explained in the section that follows.

System Boot Methods

The local-boot method uses software image and configuration files stored locally on the system. On system startup or reboot, the system looks on one of its local devices or **/hd-raid** for the specific software image and accompanying configuration text file. When using the local-booting method, you only need to configure boot stack parameters.

The system can also be configured to obtain its software image from a specific external network server while it is paired with a configuration text file that resides on the system. When using network booting, you need to configure the following:

- Boot stack parameters, which define the files to use and in what priority to use them
- Boot interface and network parameters defining the remote management LAN interface and the methods to use to reach the external network server
- Network booting delay time and optional name server parameters defining the delay period (in seconds) to allow for network communications to be established, and the IP address of any Domain Name Service (DNS) name server that may be used

Viewing the Current Boot Stack

To view the boot stack entries contained in the boot.sys file run the Exec mode **show boot** command.

Important Operator and inspector-level users can execute the **show boot** command.

The examples below shows the command output for a local booting configuration. Notice that in these examples both the image file (operating system software) and configuration file (CLI commands) are located on the **/flash** device.

Important The StarOS image filename scheme changed with release 16.1. Pre-16.1, format = "production.*image*.bin". For 16.1 onwards, format = "asr5500-*image_number*.bin". This change is reflected in the examples provided below.

Example 1 – *StarOS releases prior to 16.1*:

```
boot system priority 18 \
    image /flash/15-0-builds/production.45666.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/15-0-builds/production.45717.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/15-0-builds/production.45069.bin \
    config /flash/general_config_3665.cfg
```

Example 2 – *StarOS release 16.1 onwards*:

```
boot system priority 18 \
    image /flash/16-1-builds/asr5500-16.1.3.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/16-1-builds/asr5500-16.1.1.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/16-1-builds/asr5500-16.1.0.bin \
    config /flash/general_config_3665.cfg
```

The example below shows the output for a combination network booting and local booting configuration. Notice in this example that the first two boot stack entries (Priorities 18 and 19) load the image file (operating system software) from an external network server using the Trivial File Transfer Protocol (TFTP), while all configuration files are located on the **/flash** device.

Also notice the boot network interface and boot network configuration commands located at the top of the boot stack. These commands define what remote management LAN interface(s) to use and information about communicating with the external network server that hosts the operating system software image file.

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0
boot delay 15
boot system priority 18 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 20 image /flash/image_version.bin \config /flash/general_config.cfg
```

To identify the boot image priority that was loaded at the initial boot time enter:
show boot initial-config

The example below displays the output:

```
[local]host_name# show boot initial-config
Initial (boot time) configuration:
      image tftp://192.168.1.161/tftpboot/image_version.bin \
       config /flash/config_name.cfg
      priority 1
```

Adding a New Boot Stack Entry

Important Before performing this procedure, verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Viewing the Current Boot Stack* for more information.

If priority 1 is in use, then you must renumber the existing entry(ies) to ensure that at least that priority is available. The maximum number of boot stack entries that can be contained in the *boot.sys* file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Refer to Deleting a Boot Stack Entry, on page 8 for more information.

This procedure details how to add new boot stack entries to the *boot.sys* file. Make sure you are at the Exec mode prompt and enter the following commands:

configure
boot system priority *number* **image** *image_url* **config** *cfg_url*

The following command creates a new boot stack entry, using a boot priority of *3*.
boot system priority 3 image /flash/*image_filename***.bin config /flash/***config_name***.cfg**

Important Boot stack changes saved to the boot.sys file are not executed until the system is rebooted.

Synchronize the local file systems on the CF VMs with the following command:

filesystem synchronize all

Deleting a Boot Stack Entry

This procedure details how to remove an individual boot stack entry from the boot.sys file. Make sure you are at the Exec mode prompt and enter the following commands:

configure
no boot system priority *number*

Where *number* specifies the boot priority used for the boot stack entry. This command removes that specific entry from the boot stack, causing the boot.sys file to be overwritten.

Upgrading the Operating System Software

Identifying OS Release Version and Build Number

The operating system can be configured to provide services and perform pre-defined functions through commands issued from the CLI.

The operating system software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system.

- *For StarOS releases prior to 16.1*, the image filename is identified by its release type, build number and platform type. For example: **production.***build_number***.asr5500.bin**. For example, **production.54029.asr5500.bin**.
- *For StarOS release 16.1 onwards*, the image filename is identified by a suffix specifying its platform type and release number. For example, **asr5500-***release_number***. bin**. For example, **asr5500-16.1.0.bin**.

For StarOS releases 20.0 and higher, a starfile image must be signed with an REL key before being released. A deployable image will be signed with an REL key having a ".bin.SPA" extension, where "A" identifies the revision level of the signing key. For example, **asr5500-20.0.0.bin.SPA**. If a signing key becomes compromised, a new key is created and the revision level increments to "B".

For StarOS releases 20.0 and higher Trusted images have been introduced. The difference between a Trusted build and a Normal build is the absence of unsecure programs ftpd, telnet and tcpdump, as well as the addition of a staros.conf file for security options. Trusted images are identifiable by the presence of "_T" in the platform name. For example, **asr5500_T-20.0.0.bin.SPA**.

The software version information can be viewed from the CLI in the Exec mode by entering the **show version** command.

`[local]`*host_name*`#` **show version**

You can run the Exec mode **show build** command to display additional information about the StarOS build release.

Verify Free Space on the /flash Device

Verify that there is enough free space on the **/flash** device to accommodate the new StarOS image file by entering the following Exec mode command:
`[local]`*host_name*`#` **directory /flash**

The following is an example of the type of directory information displayed:

```
-rwxrwxr-x    1  root     root         7334 May    5 17:29 asr-config.cfg
-rwxrwxr-x    1  root     root         399 Jun    7 18:32 system.cfg
-rwxrwxr-x    1  root     root         10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x    1  root     root         10667 Jun    1 11:21 testconfig_4.cfg
-rwxrwxr-x    1  root     root         5926 Apr    7 16:27  tworpcontext.cfg
-rwxrwxr-x    1  root     root         15534 Aug    4 13:31 test_vlan.cfg
-rwxrwxr-x    1  root     root         2482 Nov 18 11:09 gateway2.cfg
-rwxrwxr-x    1  root     root         159106048 Dec 31  2011 image_filename
1136352  /flash
Filesystem           1k-blocks      Used Available Use% Mounted on
/var/run/storage/flash/part1 3115468   1136352  30018336   4% /mnt/user/.auto/onboard/flash
```

Note the "Available" blocks in the last line of the display. After displaying the directory information, the CLI returns to root and the following prompt appears:
`[local]`*host_name*`#`

Download the Software Image from the Support Site

Access to the Cisco support site and download facility is username and password controlled. You must have an active customer account to access the site and download the StarOS image.

Download the software image to a network location or physical device (USB stick) from which it can be uploaded to the **/flash** device.

Contact your Cisco representative or Cisco TAC for additional information.

Transfer StarOS Image to /flash

Transfer the new operating system image file to the /flash directory on the MIO/UMIO/MIO2 VPC-DI active CF or VPC-SI using one of the following methods:

- Transfer the file to the **/flash** device using an FTP client with access to the system.

Important Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable. In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

- Transfer the file to the **/flash** device using an SFTP client with access to the system.

Verify that the image file was successfully transferred to the /flash device by running the following Exec mode command:
`[local]`*host_name*`#` **directory /flash**

The image filename should appear in the displayed output.

Run the **show version /flash**/*image_filename* command to verify the build information.
`[local]`*host_name*`#` **show version /flash**/*image_filename***.bin**

Saving a Copy of the Current Configuration File

Prior to upgrading to a new software release, you should copy and rename the current configuration file to the **/flash** device and to an off-chassis location (external memory device or network URL). This renamed copy assures that you will have a fallback, loadable configuration file should a problem be encountered during the upgrade.

Downgrading from Release 15.0 to 14.0

Release 14 and Release 15 chassis IDs use different encryption formats. Release 15 will recognize a Release 14 chassis ID and consider it as valid. Upgrading from 14.x to 15.0 will not require changing the chassis ID or configuration file.

However, if the chassis key is reset in Release 15 through the setup wizard or **chassis-key** CLI command, a new chassis ID will be generated in Release 15 format (44 instead of 16 characters). Release 14 builds will not recognize the 44-character chassis ID. If the chassis is subsequently downgraded to Release 14, a new 16-character chassis ID will be generated. To accommodate the old key format, you must save the configuration file in pre-v12.2 format before the downgrade. If you attempt to load a v15 configuration file on the downgraded chassis, StarOS will not be able to decrypt the password/secrets stored in the configuration file.

Downgrading from Release 20.0

Prior to release 20.0, local-user passwords were hashed with the MD5 message digest-algorithm and saved in the database. In release 20. 0, PBKDF2 (Password Based Key Derivation Function - Version 2) is now used to derive a key of given length, based on entered data, salt and number of iterations. Local-user account passwords are hashed using the PBKDF2 method with a randomly generated salt coupled with a large number of iterations to make password storage more secure.

Since hash functions are one-way, it is not possible to convert PBKDF2 hashed passwords to the MD5 format. The local-user database must be downgraded prior to reverting to StarOS releases prior to 20.0.

To downgrade the local-user database to use the MD5 hash algorithm, a Security Administrator must run the Exec mode **downgrade local-user database** command. StarOS prompts for confirmation and requests the Security Administrator to reenter a password. The entered password re-authenticates the user prior to executing the downgrade command. After verification, the password is hashed using the appropriate old/weak encryption algorithm and saved in the database to allow earlier versions of StarOS to authenticate the Security Administrator.

The downgrade process does not convert PBKDF2 hashed passwords to MD5 format. The downgrade process re-reads the database (from the /flash directory), reconstructs the database in the older format, and writes it back to the disk. Since the PBKDF2 hashed passwords cannot be converted to the MD5 hash algorithm, and earlier StarOS releases cannot parse the PBKDF2 encryption algorithm, StarOS suspends all those users encrypted via the PBKDF2 algorithm. Users encrypted via the MD5 algorithm ("Weak Hash" flag) can continue to login with their credentials. After the system comes up with the earlier StarOS release, suspended users can be identified in the output of the **show local-user [verbose]**command.

To reactivate suspended users a Security Administrator can:

- Set temporary passwords for suspended users, using the Exec mode **password change local-user** *username* command.
- Reset the suspend flag for users, using the Configuration mode **no suspend local-user** *username* command.

Off-line Software Upgrade

An off-line software upgrade can be performed for any system, upgrading from any version of operating system software to any version, regardless of version number. This process is considered off-line because

while many of the steps can be performed while the system is currently supporting sessions, the last step of this process requires a reboot to actually apply the software upgrade.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:
`[local]`*host_name*`#`

Configure a Newcall Policy

Configure a newcall policy from the Exec mode to meet your service requirements. When enabled the policy redirects or rejects new calls in anticipation of the system reload that completes the upgrade process. This reduces the amount of service disruption to subscribers caused by the system reload that completes the upgrade.

Important Newcall policies are created on a per-service basis. If you have multiple services running on the chassis, you can configure multiple newcall policies.

The syntax for newcall policies is described below:
`[local]`*host_name*`#` **newcall policy { asngw-service | asnpc-service | sgsn-service } { all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy { fa-service | lns-service | mipv6ha-service } { all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy { ha-service | pdsn-service | pdsnclosedrp-service } { all | name** *service_name* **} { redirect** *target_ip_address* **[weight** *weight_num* **] [** *target_ipaddress2* **[weight** *weight_num* **] ... ***target_ip_address16* **[weight** *weight_num* **] | reject }**
`[local]`*host_name*`#` **newcall policy ggsn-service { apn name** *apn_name* **| all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy hnbgw-service { all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy { pcc-af-service | pcc-policy-service } { all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy {pcc-af-service | pcc-policy-service } { all | name** *service_name* **} reject**
`[local]`*host_name*`#` **newcall policy mme-service { all | name** *service_name* **} reject**

For complete information about the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

Configure a Message of the Day Banner

Optional: Configure a "Message of the Day" banner informing other management users that the system will be rebooted by entering the following command from the Global Configuration mode prompt.
`[local]`*host_name*`(config)#` **banner motd "***banner_text***"**

banner_text is the message that you would like to be displayed and can be up to 2048 alphanumeric characters. Note that *banner_text* must begin with and end in quotation marks (" "). For more information in entering CLI banner information, see the *CLI Reference*. The banner is displayed when an administrative user logs onto the CLI.

Back up the Current CLI Configuration File

Back up the current CLI configuration file by entering the following command:
`[local]`*host_name*`#` **copy** *from_url to_url* **[-noconfirm]**

This creates a mirror-image of the CLI configuration file linked to the operating system defined in the current boot stack entry.

The following command example creates a backup copy of a file called *general.cfg* located on the **/flash** device to a file called *general_3652.cfg*:
`[local]`*host_name*`#` **copy /flash/general.cfg /flash/general_3652.cfg**

Create a New Boot Stack Entry

Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:
`[local]`*host_name*`(config)#` **boot system priority** *number* **image** *image_url* **/flash** *filename* **config** *cfg_url* **/flash/***filename*

Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

Important Run the Exec mode **show boot** command to verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the no boot system priority command to delete a book stack entry.
`[local]`*host_name*`#` **configure**
`[local]`*host_name*`(config)#` **no boot system priority** *number*

To add new boot stack entries to the boot.sys file enter the following commands:
`[local]`*host_name*`#` **configure**
`[local]`*host_name*`(config)#` **boot system priority** *number* **image** *image_url* **config** *cfg_url*

For information on using the **boot system priority** command, refer to the Adding a New Boot Stack Entry, on page 8 .

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

To save the running configuration to the current configuration file, enter the following command:
`[local]`*host_name*`#` **save configuration /flash**

Reboot the System

Reboot the system by entering the following command:
`[local]`*host_name*`#` **reload [-noconfirm]**

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

After the system reboots, establish a CLI session and enter the **show version** command to verify that the active software version is correct.

Optional for PDSN: If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

Verify the Running Software Version

After the system has successfully booted, verify that the new StarOS version is running by executing the Exec mode **show version** command.

`[local`*host_name*`#` **show version**

You can run the Exec mode **show build** command to display additional information about the running StarOS build release.

Restoring the Previous Software Image

If for some reason you need to undo the upgrade, perform the upgrade again except:

- Specify the locations of the upgrade software image and configuration files.

then

- Specify the locations of the original software image and configuration files.

Managing License Keys

License keys define capacity limits (number of allowed subscriber sessions) and available features on your system. Adding new license keys allows you to increase capacity and add new features as your subscriber base grows.

New System License Keys

New systems are delivered with no license keys installed. In most cases, you receive the license key in electronic format (usually through e-mail).

When a system boots with no license key installed a default set of restricted session use and feature licenses is installed. The following Exec Mode command lists the license information:
`[local]`*host_name*`#` **show license information**

Important With no license key installed, the session use licenses for PDSN, HA, GGSN, and L2TP LNS are limited to 10,000 sessions.

Session Use and Feature Use Licenses

Session use and feature use licenses are software mechanisms that provide session limit controls and enable special features within the system. These electronic licenses are stored in the system's configuration file that is loaded as part of the system software each time the system is powered on or restarted.

- Session use licenses limit the number of concurrent sessions that a system is capable of supporting per service type and are acquired on an as-needed basis. This allows carriers to pay only for what they are using and easily increase capacity as their subscriber base grows.
- Feature use licenses enable specific features/functionality within the system and are distributed based on the total number of sessions supported by the system.

Installing New License Keys

Use the instructions below to install a new license key.

Cutting and Pasting the Key

If you have a copy of the license, use the following configuration to cut and paste just the license key part:

Step 1 From the Exec mode, enter the following:

configure
license key *license*
exit

license is the license key string. The license can be an alphanumeric string of 1 through 1023 characters that is case sensitive. Copy the license key as shown in the example below, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\
STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|
LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|
FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/
XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 2 Verify that the license key just entered was accepted by entering the following command at the Exec mode prompt:

`[local]`*host_name*`#` **show license key**

The new license key should be displayed. If it is not, return to the Global configuration mode and re-enter the key using the **license key** command.

Important An invalid license will not be accepted. A Failure error will appear in the output of the **license key** command when you attempt to configure an invalid license key. If you use the **-force** option to install an invalid license key, the license will be placed into a 30-day grace period. StarOS will generate daily syslog error messages and SNMP traps during the grace period. The output of the **show license information** command will indicate "License State" as "Not Valid".

Step 3 Verify that the license key enabled the correct functionality by entering the following command:

`[local]`*host_name*`#` **show license information**

All license keys and the new session capacity or functionality enabled should be listed. If the functionality or session capacity enabled by the new key is incorrect, please contact your service representative.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Caution Failure to save the new license key configuration in the current CLI configuration file will result in the loss of any of the new features enabled by the license key once the system is reloaded.

Adding License Keys to Configuration Files

License keys can be added to a new or existing configuration file.

Important License key information is maintained as part of the CLI configuration. Each time a key is installed or updated, you must re-save the configuration file.

Step 1 Open the configuration file to which the new license key commands are to be copied.

Step 2 Copy the license as shown in the example, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\STCB21M82003R80411A4|
DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|LSP=000000|LSH=000000|
LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|
SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 3 Paste the license key into the configuration

Important Paste the license key information at the beginning of the configuration file to ensure the system has the expected capacity and features before it configures contexts.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

License Expiration Behavior

When a license expires, there is a built-in grace period of 30 days that allows normal use of the licensed session use and feature use licenses. This allows you to obtain a new license without any interruption of service.

The following Exec mode command lists the license information including the date the grace period is set to expire:

show license information

Requesting License Keys

License keys for the system can be obtained through your Cisco account representative. Specific information is required before a license key may be generated:

- Sales Order or Purchase Order information
- Desired session capacity
- Desired functionality

Viewing License Information

To see the license detail, enter the following command from the Exec mode:

`[local]`*host_name*`#` **show license information [full | key [full]]**

Deleting a License Key

Use the procedure below to delete the session and feature use license key from a configuration. You must be a security administrator or administrator.

configure
no license key
exit
show license key

The output of this command should display: "No license key installed".

Managing Local-User Administrative Accounts

Unlike context-level administrative accounts which are configured via a configuration file, information for local-user administrative accounts is maintained in a separate file in flash memory and managed through the software's Shared Configuration Task (SCT). Because local-user accounts were designed to be compliant with ANSI T1.276-2003, the system provides a number of mechanisms for managing these types of administrative user accounts.

For additional information, see Disable AAA-based Authentication for Console and Limit local-user Login on Console/vty Lines.

Configuring Local-User Password Properties

Local-user account password properties are configured globally and apply to all local-user accounts. The system supports the configuration of the following password properties:

- **Complexity:** Password complexity can be forced to be compliant with ANSI T1.276-2003.
- **History length:** How many previous password versions should be tracked by the system.
- **Maximum age:** How long a user can use the same password.
- **Minimum number of characters to change:** How many characters must be changed in the password during a reset.
- **Minimum change interval:** How often a user can change their password.
- **Minimum length:** The minimum number of characters a valid password must contain.

Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details on each of the above parameters.

Configuring Local-User Account Management Properties

Local-user account management includes configuring account lockouts and user suspensions.

Local-User Account Lockouts

Local-user accounts can be administratively locked for the following reasons:

- **Login failures:** The configured maximum login failure threshold has been reached. Refer to the **local-user max-failed-logins** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details
- **Password Aging:** The configured maximum password age has been reached. Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Accounts that are locked out are inaccessible to the user until either the configured lockout time is reached (refer to the **local-user lockout-time** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference*) or a security administrator clears the lockout (refer to the **clear local-user** command in the *Exec Mode Commands* chapter of the *Command Line Interface Reference*).

Important Local-user administrative user accounts could be configured to enforce or reject lockouts. Refer to the **local-user username** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Local-User Account Suspensions

Local-user accounts can be suspended as follows:

configure
suspend local-user *name*

A suspension can be removed by entering:
configure
no suspend local-user *name*

Changing Local-User Passwords

Local-user administrative users can change their passwords using the **password change** command in the Exec mode. Users are prompted to enter their current and new passwords.

Security administrators can reset passwords for local-users by entering the following command from the root prompt in the Exec mode:
`[local]`*host_name*`#` **password change username** *name*

name is the name of the local-user account for which the password is to be changed. When a security administrator resets a local-user's password, the system prompts the user to change their password the next time they login.

All new passwords must adhere to the password properties configured for the system.