BROCADE

Brocade SLX-OS Management Configuration Guide, 16r.1.01

Supporting the Brocade SLX 9850 Router

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional.
	Default responses to system prompts are enclosed in square brackets.
{ x | y | z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
	In Fibre Channel products, square brackets may be used instead for this purpose.
x | y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.

Click the **Support** tab and select **Document Library** to access product documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

* Through the online feedback form in the HTML documents posted on www.brocade.com
* By sending your feedback to documentation@brocade.com

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online or by telephone. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.

- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.

- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.

- For questions regarding service levels and response times, contact your OEM/solution provider.

About This Document

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc. for SLX-OS Release 16r.1.01, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release:

* Brocade SLX 9850-4 router
* Brocade SLX 9850-8 router

To obtain information about other Brocade OS versions, refer to the documentation specific to that version.

Configuration Fundamentals

Configuration files

Brocade devices support three types of configuration files, default, startup, and running configuration files.

The startup configuration resides in the /var/config/vcs/scripts directory. Though the default configuration files are physically present in this directory, they are linked to the directory from their actual location.

When you boot up a device for the first time, the running configuration is identical to the startup configuration. As you configure the device, the changes are written to the running configuration. To save the changes, you must save the currently effective configuration (the running configuration) as the startup configuration. When the device reboots, the configuration changes become effective.

Default configuration files

Default configuration files are part of the firmware package for the device and are automatically applied to the startup configuration under the following conditions:

- When the device boots up for the first time and no customized configuration is available.
- When you restore the default configuration.

You cannot remove, rename, or change the default configuration. The default configuration file names are as follows:

- defaultconfig.standalone
- defaultconfig.cluster

Startup configuration files

The startup configuration is persistent. It is applied when the system reboots.

- When the device boots up for the first time, it uses the default configuration as the startup configuration, depending on the mode.
- When you make configuration changes to the running configuration and save the changes to the startup configuration with the **copy** command, the running configuration becomes the startup configuration.

The startup configuration file name is startup-config.

Running configuration files

The configuration currently effective on the device is referred to as the running configuration. Any configuration change you make while the device is online is made to the running configuration.

- The running configuration is nonpersistent.
- To save configuration changes, you must copy the running configuration to the startup configuration. If you are not sure about the changes, you can copy the changes to a file, and apply the changes later.

The running configuration file name is running-config.

Displaying configurations

The following examples illustrate how to display the default, startup, and running configurations, respectively.

Displaying the default configuration

To display the default configuration, enter the **show file** command with the default configuration filenames in privileged EXEC mode.

```
device# show file defaultconfig.standalone
device# show file defaultconfig.cluster
```

Displaying the startup configuration

To display the contents of the startup configuration, enter the **show startup-config** command in privileged EXEC mode.

```
device# show startup-config
```

Displaying the running configuration

To display the contents of the running configuration, enter the **show running-config** command in the privileged EXEC mode.

```
device# show running-config
```

Saving configuration changes

Configuration changes are nonpersistent and are lost on reboot unless you save them permanently. You have two options for saving configuration changes:

- Copy the running configuration to the startup configuration. The changes become effective upon reboot.
- Copy the running configuration to a file, and apply it at some later date.

  NOTE
  Always make a backup copy of your running configuration before you upgrade or downgrade the firmware.

Saving the running configuration

To save the configuration changes you made, copy the running configuration to the startup configuration. The next time the device reboots, it uses the startup configuration and the changes you made earlier become effective.

Enter the **copy running-config startup-config** command in privileged EXEC mode.

```
device# copy running-config startup-config
This operation will modify your startup configuration. Do you want to continue? [Y/N]: y
```

Saving the running configuration to a file

If you want to save the changes you made to the configuration, but you do not want the changes to take effect when the device reboots, you can save the running configuration to a file. You can apply the changes at some later time.

Enter the **copy running-config** command in privileged EXEC mode. Specify the file name as the file URL.

```
device# copy running-config flash://myconfig
```

Verify the transaction by listing the directory contents.

```
device# dir
total 32
drwxr-xr-x 2  251 1011   4096 Jun  5 07:08 .
drwxr-xr-x 3  251 1011   4096 Mar 11 00:00 ..
-rw-r--r-- 1 root sys     410 Jun  3 00:56 defaultconfig.standalone
-rw-r--r-- 1 root sys     695 Jun  3 00:56 defaultconfig.cluster
-rw-r--r-- 1 root root 183118 Jun  7 12:19 myconfig
-rw-r--r-- 1 root root 185650 Jun  5 09:38 startup-config
```

Applying previously saved configuration changes

When you are ready to apply the configuration changes you previously saved to a file, copy the file (*myconfig* in the example) to the startup configuration. The changes take effect after the device reboots.

Enter the **copy** command in privileged EXEC mode. Specify the file name as the file URL followed by the **startup-config** keyword.

```
device# copy flash://myconfig startup-config
This operation will modify your startup configuration. Do you want to continue? [Y/N]: y
```

Backing up configurations

Always keep a backup copy of your configuration files, so you can restore the configuration in the event the configuration is lost or you make unintentional changes.

The following recommendations apply:

*   Upload the configuration backup copies to an external host or to an attached Brocade-branded USB device.

    > NOTE
    > USB access is always to the locally active management module (MM).

*   Avoid copying configuration files from one device to another. Instead restore the device configuration files from the backup copy.

Copying a configuration file to an external host

You can copy the startup-config or running-config file to a remote server through FTP, SCP, TFTP, or SFTP.

In the following example, the startup configuration is copied to a file on a remote server by means of FTP.

```
device# copy startup-config ftp://admin:******@10.34.98.133//archive/startup-config_device24-08_20101010
```

Backing up the startup configuration to a USB device

When you make a backup copy of a configuration file on an attached USB device, the destination file is the file URL on the USB device. You do not need to specify the target directory. The file is automatically recognized as a configuration file and stored in the default configuration directory.

1. Enable the USB device.

    ```
    device# usb on
    USB storage enabled
    ```

2. Enter the **copy startup-config** command with the destination file name.

    ```
    device# copy startup-config usb://startup-config_slx-08_20160510
    ```

Configuration restoration

Restoring a configuration involves overwriting a given configuration file on the device by downloading an archived backup copy from an external host or from an attached USB device.

All interfaces remain online. The following parameters are unaffected:

*   Interface management IP address
*   Software feature licenses installed on the device
*   Virtual IP address

    NOTE
    USB access is always to the locally active management module (MM).

Restoring the default configuration

This restoration procedure resets the configuration to the factory defaults. The default configuration files are always present on the device and can be restored with the **copy** command.

To restore the default configuration, perform the following procedure in privileged EXEC mode.

1. Enter the **copy default-config startup-config** command to overwrite the running configuration with the default configuration.

    ```
    device# copy default-config startup-config
    ```

2. Confirm that you want to make the change by entering Y when prompted.

    ```
    This operation will modify your startup configuration. Do you want to continue? [Y/N]: y
    ```

3. Reboot the device.

    ```
    device# reload system
    ```

Managing flash files

The Brocade device provides a set of tools for removing, renaming, and displaying files you create in the device flash memory. You can use the display commands with any file, including the system configuration files. The **rename** and **delete** commands only apply to copies of configuration files you create in the flash memory. You cannot rename or delete any of the system configuration files.

Listing the contents of the flash memory

To list the contents of the flash memory, enter the **dir** command in privileged EXEC mode.

```
device# dir
total 572
drwxr-xr-x 2  251 1011   4096 Jun  5 07:08 .
drwxr-xr-x 3  251 1011   4096 Mar 11 00:00 ..
-rw-r--r-- 1 root sys     410 Jun  3 00:56 defaultconfig.standalone
-rw-r--r-- 1 root sys     695 Jun  3 00:56 defaultconfig.cluster
-rw-r--r-- 1 root root 185650 Jun  5 09:38 startup-config
```

Deleting a file from the flash memory

To delete a file from the flash memory, enter the **delete** command with the file name in privileged EXEC mode.

```
device# delete myconfig
```

> **NOTE**
> You cannot delete a system configuration file in flash memory.

Renaming a flash memory file

To rename a file in the flash memory, enter the **rename** command with the source and destination file names in privileged EXEC mode.

```
device# rename myconfig myconfig_20101010
```

> **NOTE**
> You cannot rename a system configuration file in flash memory.

Viewing the contents of a file in the flash memory

To investigate the contents of a file in the flash memory, enter the **show file** command with the file name in privileged EXEC mode.

```
device# show file defaultconfig.cluster
vlan dot1q tag native
!
cee-map default
remap fabric-priority priority 0
remap lossless-priority priority 0
priority-group-table 15.0 pfc off
priority-group-table 1 weight 40 pfc on
priority-group-table 2 weight 60 pfc off
priority-table 2 2 2 1 2 2 2 15.0
!
!
port-profile default
vlan-profile
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
!
interface Port-channel 1
  vlag ignore-split
  description Homerun port-channel on MM1
  shutdown
!
interface Port-channel 2
  vlag ignore-split
  description Homerun port-channel on MM2
  shutdown
!
protocol lldp
!
```

```
!
logging auditlog class CONFIGURATION
logging auditlog class FIRMWARE
logging auditlog class SECURITY
!
end
```

> **NOTE**
> To display the contents of the running configuration, use the **show running-config** command. To display the contents of the startup configuration, use the **show startup-config** command.

Rebooting the device

You can reboot the management module or the chassis.

> ⚠ **CAUTION**
> **All reboot operations are disruptive, and the commands prompt for confirmation before executing. When you reboot a device, all traffic to and from it stops. All ports on that device remain inactive until the device comes back online.**

> **NOTE**
> During the boot process system initialization, configuration data (default or user-defined) are applied to the device through configuration replay.

- The **fastboot** commands only reboots the management module on which the command is executed. Any unsaved configurations are lost. If you log in to the device IP address and enter this command, only the active management module reboots and POST is bypassed.

  ```
  device# fastboot
  ```

- The **reload system** command performs a cold reboot that powers off and restarts the entire chassis. All session connections must be restarted. If the power-on self-test (POST) is enabled, POST is executed when the system comes back up.

  ```
  device# reload system
  ```

Session connection

You can connect to your device through a console session on the serial port, or through a Telnet or Secure Shell (SSH) connection to the management port or the inband port belonging to either the mgmt-vrf or default-vrf Virtual Routing and Forwarding (VRF). You can use any account login present in the local device database or on a configured authentication, authorization, and accounting (AAA) server for authentication. For initial setup procedures, use the pre-configured administrative account that is part of the default device configuration.

The device must be physically connected to the network. If the device network interface is not configured or the device has been disconnected from the network, use a console session on the serial port.

Refer to the appropriate hardware guide for information on connecting through the serial port and establishing an Ethernet connection for a console session.

Telnet

Telnet allows access to management functions on a remote networking device. Unlike SSH, Telnet does not provide a secure, encrypted connection to the device.

Telnet support is available in privileged EXEC mode on all Brocade platforms. The device supports a maximum of 32 CLI sessions. Both IPv4 and IPv6 addresses are supported.

The Telnet service is enabled by default on the device. When the Telnet server is disabled, remote access to the device is restricted. Existing inbound Telnet connections are terminated and access to the device by additional inbound connections is not allowed until the Telnet server is re-enabled. If you have admin privileges, you can disable and re-enable inbound Telnet connections from global configuration mode.

> **NOTE**
> Outgoing Telnet connections from the device to any remote device are not affected by disabling or enabling the Telnet server in the device.

> **NOTE**
> When using Telnet, the root ID is blocked and you cannot login as root.

The following features are not supported with Telnet:

- Displaying Telnet sessions
- Terminating hung Telnet sessions

Connecting to a Brocade device with Telnet

You can use the Telnet service to connect to the Brocade device.

A Telnet session allows you to access a device remotely using port 23. However, it is not secure. If you need a secure connection, use SSH.

1. Establish a Telnet session to the Brocade device from a remote device.

   ```
   client# telnet 10.17.37.157
   ```

   The example establishes a Telnet session to the device with the IP address of 10.17.37.157.

   If the device is active and the Telnet service is enabled on it, a display similar to the following appears.

   ```
   Trying 10.17.37.157...
   Connected to 10.17.37.157.
   Escape character is '^]'.
   ```

2. Once you have established the Telnet connection, you can log in normally.

   ```
   device login: admin
   Password:
   SECURITY WARNING: The default password for at least
   one default account (root, admin and user) have not been changed.

   Welcome to the Brocade SLX-OS Software
   admin connected from 10.252.24.5 using telnet on device
   device#
   ```

   > **NOTE**
   > The default admin login name is admin. The default user name is user. The default password for both admin and user accounts is password.

Brocade recommends that you change the default account password when you log in for the first time. For more information on changing the default password, refer to the *Brocade SLX-OS Security Configuration Guide*.

Connecting to a remote device with Telnet

You can connect to a remote server from the device using a Telnet connection.

A Telnet session is not secure. If you need a secure connection, use SSH.

To connect to a remote server with Telnet, perform the following steps:

1. Establish a Telnet session connection to the remote device.

    ```
    device# telnet 10.20.51.68 vrf mgmt-vrf
    ```

    The example establishes a Telnet session to a device with the IP address of 10.20.51.68.

    You can override the default port by using the **port-number** *port* option. However, the device must be listening on this port for the connection to succeed.

    If the device is active and the Telnet service is enabled on it, a display similar to the following appears.

    ```
    /fabos/cliexec/vrfcmd 0 /usr/bin/telnet 10.20.51.68 23
    Trying 10.20.51.68...
    Connected to 10.20.51.68.
    Escape character is '^]'.
    ...
    device login:
    ```

2. Once you have established the Telnet connection, you can log in normally.

Shutting down and re-enabling the Telnet service

The Telnet service is enabled by default. Shutting down the Telnet service forcibly disconnects all Telnet sessions running on a device.

To shut down and then re-enable the Telnet service, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

    ```
    device# configure terminal
    ```

2. To shut down the Telnet service on the device, enter **telnet server shutdown**.

    ```
    device(config)# telnet server shutdown
    ```

    All Telnet sessions including any currently active sessions are immediately terminated, and cannot be re-established until the service is re-enabled.

3. To re-enable the Telnet service on the device, enter **no telnet server shutdown**.

    ```
    device(config)# no telnet server shutdown
    ```

SSH

Secure Shell (SSH) allows secure access to management functions on a remote networking device. Unlike Telnet, which offers no security, SSH provides a secure, encrypted connection to the device.

SSH support is available in privileged EXEC mode on all Brocade platforms. The device supports a amximum of 32 CLI sessions. Both IPv4 and IPv6 addresses are supported.

The SSH service is enabled by default on the device. When the SSH server is disabled, remote access to the device is restricted. Existing inbound SSH connections are terminated and access to the device by additional inbound connections are not allowed until the SSH

server is re-enabled. If you have admin privileges, you can disable and re-enable inbound SSH connections from global configuration mode.

> **NOTE**
> Outgoing SSH connections from the device to any remote device are not affected by disabling or enabling the SSH server in the device.

> **NOTE**
> When using SSH, the root ID is blocked and you cannot login as root.

Feature support for SSH

SSHv2 is the supported version of SSH, but not all features typically available with SSHv2 are supported on the Brocade devices.

The following encryption algorithms are supported:

- **3des-cbc** Triple-DES (default)
- **aes256-cbc** : AES in Cipher Block Chaining (CBC) mode with 256-bit key
- **aes192-cbc** : AES in CBC mode with 192-bit key
- **aes128-cbc** : AES in CBC mode with 128-bit key
- **aes256-gcm** : AES in Galios/Counter Mode (GCM) mode with 256-bit key
- **aes192-gcm** : AES in GCM mode with 192-bit key
- **aes128-gcm** : AES in GCM mode with 128-bit key
- **aes256-ctr** : AES in Counter Mode (CTR) mode with 256-bit key
- **aes192-ctr** : AES in CTR mode with 192-bit key
- **aes128-ctr** : AES in CTR mode with 128-bit key

The following Hash-based Message Authentication Code (HMAC) message authentication algorithms are supported:

- **hmac-md5** : MD5 encryption algorithm with 128-bit key (default).
- **hmac-sha1** : SHA1 encryption algorithm with 160-bit key.
- **hmac-sha2-256** : SHA2 encryption algorithm with 256-bit key.
- **hmac-sha2-512** : SHA2 encryption algorithm with 512-bit key.

The following host keys are supported:

- ssh-dsa
- ssh-rsa
- ECDSA

The following key exchange algorithms are supported:

- diffie-hellman-group-exchange-sha256
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group14-sha1
- diffie-hellman-group1-sha1

SSH user authentication is performed with passwords stored on the device or on an external authentication, authorization, and accounting (AAA) server.

The following features are not supported with SSH:

- Displaying SSH sessions

- Deleting stale SSH keys

Connecting to a Brocade device with SSH

You can use SSH to connect to the Brocade device.

An SSH session allows you to access a device remotely using port 22.

1. Establish an SSH session connection to the Brocade device.

```
client# ssh admin@10.17.37.157
```

The example establishes an SSH session to the device with the IP address of 10.17.37.157.

2. Enter yes if prompted.

```
The authenticity of host '10.17.37.157 (10.17.37.157)' can't be established.
RSA key fingerprint is 9f:83:62:cd:55:6c:b9:e8:1d:79:ab:b4:04:f4:f6:2a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.17.37.157' (RSA) to the list of known hosts.
admin@10.17.37.157's password:


SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.

Welcome to the Brocade SLX-OS Software
admin connected from 10.70.4.113 using ssh on device
device#
```

> **NOTE**
> The default admin login name is admin. The default user name is user. The default password for both admin and user accounts is password.

Brocade recommends that you change the default account password when you log in for the first time. For more information on changing the default password, refer to the *Brocade SLX-OS Security Configuration Guide*.

Connecting to a remote server with SSH

You can connect to a remote server from the device using the SSH (Secure Socket Handling) protocol to permit a secure (encrypted) connection.

To connect to a remote server with SSH, perform the following steps:

1. Establish an SSH connection with the login name and IP address for the remote server.

```
device# ssh 10.20.51.68 -l admin vrf mgmt-vrf
```

You can use the **-m** and **-c** options to override the default encryption and hash algorithms

2. Enter **yes** if prompted.

```
The authenticity of host '10.20.51.68 (10.20.51.68)' can't be established.
RSA key fingerprint is ea:32:38:f7:76:b7:7d:23:dd:a7:25:99:e7:50:87:d0.
Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.20.51.68' (RSA) to the list of known hosts.
admin@10.20.51.68's password: ********

SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.
Welcome to the Brocade SLX-OS Software
admin connected from 10.20.51.66 using ssh on C60_68F
```

Managing SSH public keys

You can import SSH public keys to establish an authenticated login for a device. You can also delete the key from the device to prevent it from being used for an authenticated login.

To manage the SSH keys, perform the following steps:

1. In privileged EXEC mode, import an SSH public key to the device.

    ```
    device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh
    file id_rsa.pub login fvt
    ```

    This example imports the SSH public key for the admin user from the remote 10.70.4.106 host using the directory and file information to the key.

2. Enter the password for the user.

    ```
    Password: ***********
    ```

    When the SSH key is imported, the following message appears.

    ```
    device# 2016/01/14-10:28:58, [SEC-3050], 75,, INFO, SLX9850-4, Event: sshutil, Status: success,
    Info: Imported SSH public key from 10.70.4.106 for user 'admin'.
    ```

3. Delete an SSH public key from the device prevents it from being used.

    ```
    device# no certutil sshkey user admin
    ```

    This example deletes the SSH key for the admin user.

Shutting down and re-enabling the SSH service

The SSH service is enabled by default. Shutting down the SSH service forcibly disconnects all SSH sessions running on a device.

To shut down and then re-enable the SSH service, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Shut down the SSH service on the device.

    ```
    device(config)# ssh server shutdown
    ```

    All SSH sessions are immediately terminated, and cannot be re-established until the service is re-enabled.

3. To re-enable the SSH service on the device, enter **no ssh server shutdown**.

    ```
    device(config)# no ssh server shutdown
    ```

Configuring the terminal session parameters

You can set the terminal parameters for the current session. You can also set password attributes for the length of the session and login attempts.

To set the parameters, perform the following steps:

1. In privileged EXEC mode, set the display length.

    ```
    device# terminal length 30
    ```

    This example sets the lines to be displayed on the terminal session at 30 lines.

2. Set the timeout length.

```
device# terminal timeout 60
```

This example sets the timeout of 60 minutes for the terminal session.

3. Access global configuration mode.

```
device# configure terminal
```

4. Configure the maximum login attempts to establish a session.

```
device(config)# password-attributes max-retry 4
```

This example sets the maximum login attempts of four to establish a session.

5. Set the maximum number of minutes after which the user account is unlocked.

```
device(config)# password-attributes admin-lockout max-lockout-duration 5
```

This example specifies that the user account be unlocked after 5 minutes.

The following configuration is the example of the previous steps.

```
device# terminal length 30
device# terminal timeout 60
device# configure terminal
device(config)# password-attributes max-retry 4
device(config)# password-attributes admin-lockout max-lockout-duration 5
```

Configuring a login banner

The Brocade device can be configured to display a greeting message on user terminals as a banner when they enter the Privileged EXEC CLI level or access the device through Telnet.

Complete the following steps to set and display a banner.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

2. Configure the login banner.

```
device(config)# banner login "Please do not disturb the setup on this device"
```

This example configure a text message on a single line by enclosing the text in double quotation marks (" ").

The banner can be up to 2048 characters long. To create a multi-line banner, enter the **banner login** command followed by the **Esc-m** keys. Enter **Ctrl-D** to terminate the input.

You can use the **no banner login** command to remove the banner.

3. Verify the configured banner.

```
device(config)# do show running-config banner
```

The configured banner is displayed.

```
banner login "Please do not disturb the setup on this device"
```

The following example is the configuration of the previous steps.

```
device# configure terminal
Entering configuration mode terminal
device(config)# banner login "Please do not disturb the setup on this device"
```

Ethernet management interfaces

The Ethernet network interface provides management access, including direct access to the device CLI. You must configure at least one IP address using a serial connection to the CLI before you can manage the system with other management interfaces. You can either configure static IP addresses, or you can use a Dynamic Host Configuration Protocol (DHCP) client to acquire IP addresses automatically. For IPv6 addresses, both static IPv6 and stateless IPv6 autoconfiguration are supported.

ATTENTION
Setting static IPv4 addresses and using DHCP are mutually exclusive. If DHCP is enabled, remove the DHCP client before you configure a static IPv4 address. However, this does not apply to IPv6 addresses.

Configuring a static Ethernet address

You can configure IPv4 and IPv6 static Ethernet network interface addresses in environments where the DHCP service is not available.

Before you configure a static address, connect to the device through the serial console.

To configure static Ethernet network interface addresses, perform the following steps:

1. In privileged EXEC mode, enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Access the interface management mode for the management interface.

    ```
    device(config)# interface Management 1
    ```

    This interface uses the default mgmt-vrf VRF.

3. Disable DHCP.

    ```
    device(config-Management-1)# no ip address dhcp
    ```

4. Configure the IP address for the management interface.

    ```
    device(config-Management-1)# ip address 10.24.85.81/20
    ```

5. If you are going to use an IPv6 address, configure the address.

    ```
    device(config-Management-1)# ipv6 address fd00:60:69bc:832:e61f:13ff:fe67:4b94/64
    ```

6. Verify the configuration.

    ```
    device(config-Management-1)# do show running-config interface Management 1
    interface Management 1
     no ip address dhcp
     ip address 10.24.85.81/20
     ipv6 address fd00:60:69bc:832:e61f:13ff:fe67:4b94/64
    !
    ```

The following example is the configuration of the previous steps.

```
device# configure terminal
device(config)# interface Management 1
device(config-Management-1)# no ip address dhcp
device(config-Management-1)# ip address 10.24.85.81/20
device(config-Management-1)# ipv6 address fd00:60:69bc:832:e61f:13ff:fe67:4b94/64
```

Displaying a management interface

You can display the information about a management interface on the device. If an IP address has not been assigned to the network interface, you must connect to the CLI using a console session on the serial port. Otherwise, connect to the device through Telnet or SSH.

```
device# show interface Management 1
interface Management 1
 line-speed actual "1000baseT, Duplex: Full"
 line-speed configured Auto
 oper-status up
 ip address "static 10.20.161.66/20"
 ipv6 ipv6-address [ "static 2620:100:0:f814:10:20:161:66/64 preferred" ]
```

Port management

The Brocade device allows the port management of the Brocade SLX 9850 36x100G line card (LC), Brocade SLX 9850 72X10G line card, and features for interface Ethernet ports.

- Brocade SLX 9850 36x100G line card (LC) port management includes following:
    - 40G break-out to configure any 40G port in 36x100G LC as four 10G ports
    - 40G or 100G dual-speed port groups
- Brocade SLX 9850 72X10G line card port management allows you to configure its port speed to 1G or 10G and negotiation.
- Interface Ethernet port management features discussed this section include the following:
    - Port transition hold timer
    - Port flap dampening
    - Link fault signaling

Brocade SLX 9850 36x100G LC ports

The Brocade SLX 9850 36x100G line card (LC) has six port groups with 10 ports each for a total of 60 ports. By default, each port group has six ports that are configured for 100G. The remaining ports are inactive.

On a port group, you can configure dual speed and breakout mode support.

- Dual speed support allows you to configure the ports in a port group as 40G, supporting QSFP+ optics. With this configuration, all ports in the group become active.
- Breakout mode support allows you to break out each 40G port into four 10G ports.

The following example displays the slot information for all slots in the chassis. The Brocade SLX 9850 36x100G line cards are in slots L1, L3, and L4.

```
device# show slots
Slot   Type         Description              ID      Status
------------------------------------------------------------------
 M1    MM v5        Management Module        179     ENABLED
 M2                                                  VACANT
```

```
S1                                               VACANT
S2   SFM4 v4      Switch Fabric Module    183    ENABLED
S3                                               VACANT
S4   SFM4 v4      Switch Fabric Module    183    ENABLED
S5   SFM4 v4      Switch Fabric Module    183    ENABLED
S6                                               VACANT
L1   LC36x100G v3 36-port 100GE card      181    ENABLED
L2   LC72X10G v5  72-port 10GE card       180    ENABLED
L3   LC36x100G v3 36-port 100GE card      181    ENABLED
L4   LC36x100G v3 36-port 100GE card      181    ENABLED
```

Limitations and considerations

- This LC can support 10G, 40G and 100G port speeds.

- Ports 1-6, 8-13, 15-20, 41-46, 48-53, and 55-60 can be configured as 100G ports.

- Ports 7, 14, 21-40, 47, 54 are 40G ports only.

- Both dual speed and breakout mode changes require the reload of the line card.

Configuring dual-speed ports

By default, six ports of a 36X100G port group are configured as 100G and the remaining four are inactive. You can configure the ports in a port group as 40G and activate all 10 ports.

Before performing the following procedure, you can display the port groups and their ports by using the **show hardware port-group** command.

```
device# show hardware port-group
Port-Group               Port
=============================================================
3/1          3/1  ,2  ,3  ,4  ,5  ,6  ,7  ,26 ,27 ,28
3/2          3/8  ,9  ,10 ,11 ,12 ,13 ,29 ,30 ,31 ,32
3/3          3/15 ,16 ,17 ,18 ,19 ,20 ,14 ,33 ,34 ,35
3/4          3/41 ,42 ,43 ,44 ,45 ,46 ,21 ,22 ,23 ,24
3/5          3/48 ,49 ,50 ,51 ,52 ,53 ,25 ,36 ,47 ,54
3/6          3/55 ,56 ,57 ,58 ,59 ,60 ,37 ,38 ,39 ,40

* Only the first six ports in each group are 100g capable ports
```

Perform the following steps to configure the ports in a port group:

1. From privileged EXEC mode, power off the line card.

   ```
   device# power-off linecard 3
   ```

   Before changing the port speed, you must power off the LC.

   > **NOTE**
   > The interface and its current configuration will still exist in the configuration, displayed by the **show running-config** command, but operational commands do not display interfaces on this line card.

2. Access global configuration mode.

   ```
   device# configure terminal
   ```

3. Access hardware configuration mode.

   ```
   device(config)# hardware
   ```

4. Access the port group mode for the port group that you want to configure.

```
device(config-hardware)# port-group 3/1
```

5. Configure the speed of the ports in the group to 40G.

```
device(config-port-group-3/1)# mode 40g
%Warning: port-group mode is a disruptive command.
Please power-cycle the line-card for the changes to take place
```

To reset the port speed to 100G, use the **mode 100g** command.

6. Access privileged EXEC mode.

```
device(config-port-group-3/1)# Ctrl-z
```

> **NOTE**
> You can configure more than one port group on a LC before restoring its power.

7. Restore power to the line card.

```
device# power-on linecard 3
```

8. Verify the configuration.

```
device# show running-config hardware port-group
hardware
port-group 3/1
  mode 40g
 !
```

The following example shows the steps of the previous configuration.

```
device# power-off linecard 3
device# configure terminal
device(config)# hardware
device(config-hardware)# port-group 3/1
device(config-port-group-3/1)# mode 40g
%Warning: port-group mode is a disruptive command.
Please power-cycle the line-card for the changes to take place
device(config-port-group-3/1)# Ctrl-z
device# power-on linecard 3
```

Configuring breakout mode on a 40G port

You can configure any 40G port on the 36X100G LC as four 10G ports.

Before performing the following procedure, you can view the ports in the chassis and their status, use the **show interface brief** command:

```
Interface                 IP-Address Vrf              Status               Protocol
========================= ========== ================ ==================== ========
Port-channel 1            unassigned                  administratively down down
Port-channel 2            unassigned                  administratively down down
Ethernet 3/1              unassigned default-vrf      up                   up
...
Ethernet 3/72             unassigned default-vrf      up                   down
Ethernet 3/125            unassigned default-vrf      up                   down
Ethernet 3/126            unassigned default-vrf      up                   down
```

Perform the following steps:

1. In privileged EXEC mode, power off the appropriate line card.

   ```
   device# power-off linecard 4
   ```

   > **NOTE**
   > The interface and its current configuration will still exist in the configuration, displayed by the **show running-config** command, but operational commands will not display interfaces on this line card.

2. Access global configuration mode.

   ```
   device# configure terminal
   ```

3. Access hardware configuration mode.

   ```
   device(config)# hardware
   ```

4. Access the connector configuration mode for the port you wish to break out.

   ```
   device(config-hardware)# connector 4/2
   ```

5. Configure the port into four 10G ports.

   ```
   device(config-connector-4/2)# breakout mode 4x10g
   %Warning: Sfp Breakout is a disruptive command.
   Please save the running-config to startup-config and use reload command on the device or power-cycle
   linecard on Chassis system for the changes to take effect.
   ```

6. Access privileged EXEC mode.

   ```
   device(config-connector-4/2)#Ctrl-z
   ```

7. Restore power to the line card.

   ```
   device# power-on linecard 4
   ```

   > **NOTE**
   > The SFP interfaces come up under the new mode with default configurations. Unaffected interfaces retain the configurations they had before the line card was powered off.

8. Verify the configuration.

   ```
   device# show running-config hardware connector
   hardware
    connector 4/2
      breakout mode 4x10g
    !
   ```

The following example shows the steps of the previous configuration.

```
device# power-off linecard 4
device# configure terminal
device(config)# hardware
device(config-hardware)# connector 4/2
device(config-connector-4/2)# breakout mode 4x10g
%Warning: Sfp Breakout is a disruptive command.
Please save the running-config to startup-config and use reload command on the device or power-cycle
linecard on Chassis system for the changes to take effect.
device(config-port-group-4/2)# Ctrl-z
device# power-on linecard 4
```

10G/1G auto negotiation and auto detection mode

Only the Brocade SLX 9850 72X10G line card supports 10G/1G auto negotiation and auto detection mode.

- Auto negotiation is supported on ports 1 to 24 on the front plate. However, ports 25 to 72 support 1G mode without auto negotiation.
- Auto detection occurs when the interface speed is configured based on the detected optic type.
- Only full duplex 10G or 1G mode is supported.

You can manually configure the port speed to 1G or 10G. In manual 1G or 10G mode, the inserted optic must match the configured speed. Otherwise, the link will not come up. You can configure 1G mode with or without auto negotiation.

Configuring port speed and negotiation

You can configure the speed and negotiation for Brocade SLX 9850 72X10G line card ports.

In manual 1G or 10G mode, the inserted optic must match the configured speed.

Perform the following steps to change the port speed.

1. From privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access the interface configuration mode for the port that you want to configure.

   ```
   device(config)# interface Ethernet 2/1
   ```

3. Change the port speed of interface, if required.

   ```
   device(config-if-eth-2/1# speed 1000-auto
   ```

   In this example, the port speed of interface 2/1 is changed from the default of 10G/1G auto detection mode to 1G with auto negotiation.

4. Activate the interface, if required.

   ```
   device(config-if-eth-2/1)# no shutdown
   ```

5. Verify the configuration.

   ```
   device(config-if-eth-2/1)# do show running-config interface ethernet 1/8
   interface Ethernet 2/1
    ....
    speed 1000-auto
    no shutdown
   !
   ```

The following example shows the steps in the previous configuration.

```
device# configure terminal
device(config)# interface Ethernet 2/1
device(config-if-eth-2/1)# speed 1000-auto
device(config-if-eth-2/1)# no shutdown
```

Port flap dampening

Port flap dampening allows you to configure a wait period before a port, whose link goes down then up, becomes enabled.

If the port link state toggles, from down to up or from up to down, for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port's link state is re-enabled. However, if the wait period is set to zero (0) seconds, or you want to re-enable the port before the wait period expires, the port must be manually re-enabled.

Configuring port flap dampening

By default, port flap dampening is disabled on the device. You can configure the threshold of link flapping to shut down the port and the time interval in which it remains shut down. This feature is available for all front ports on the device and is configured on the interface level.

Perform the following steps to configure port flap dampening:

1. From privileged EXEC mode, access global configuration mode.

    ```
    device# configure terminal
    ```

2. Access the interface configuration mode for the port that you want to configure.

    ```
    device(config)# interface Ethernet 1/4
    ```

3. Configure port flap dampening.

    ```
    device(conf-if-eth-1/4)# link-error-disable 10 3 10
    ```

    In this example, the values for the parameters are as follows:

    *   The toggle threshold is set to 10 times. The threshold is the number of times that the port's link state goes from up to down and down to up before the wait period is activated.

    *   The sampling time is set to 3 seconds. This time period is the amount of time during which the specified toggle threshold can occur before the wait period is activated.

    *   The wait time is set to 10 seconds. This period of time is the amount of time the port remains disabled (down) before it becomes enabled. Entering 0 indicates that the port will stay down until an administrative override occurs.

The following example shows the steps in the previous configuration.

```
device# configure terminal
device(config)# interface Ethernet 1/4
device(conf-if-eth-1/4)# link-error-disable 10 3 10
```

Port transition hold timer

The port transition hold timer delays the sending of port up or down events to Layer 2 protocols and prevents port link flapping from affecting upper layer protocols or applications. After the delay expires, the event is sent to the upper layer.

While link down events are reported immediately in the Syslog, their effect on higher level protocols such as OSPF is delayed according to how the hold timer is configured. When configured, the timer affects the physical link events. However, the resulting logical link events are also delayed.

> **NOTE**
> All LAG member ports must have the same delayed-link-event configuration.

> **NOTE**
> The delayed-link-event configuration is applicable only on a physical interface. It is not valid on a VLAN, VE, LAG, or loopback interfaces.

NOTE
The port transition hold timer does not take effect when the interface is administratively shut down.

Configuring the port transition hold timer

By default, the sending of an up or down port event is not delayed. You can configure a delay for either or both events.

Perform the following steps to configure the port transition hold timer:

1. From privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access the interface configuration mode for the port that you want to configure.

   ```
   device(config)# interface Ethernet 4/2
   ```

3. Configure the port transition hold timer.

   ```
   device(conf-if-eth-4/2)# delay-link-event 2 down
   ```

   The polling iteration is 50 ms. In this example, 50 ms is multiplied by 2 and the sending of port down event is delayed by 100 ms. If the port is detected to be in the up state within the 100 ms, the delayed down event is cancelled.

   You can specify a multiplier value from 1 to 200 for delay times from 50 ms to 10 seconds and a port event of **up**, **down**, or **both**.

4. Verify the configuration.

   ```
   device(conf-if-eth-4/2)# do show running-config internet ethernet 4/2
   interface Ethernet 4/2
    ...
    delay-link-event 2 down
    no shutdown
   !
   ```

The following example shows the steps in the previous configuration.

```
device# configure terminal
device(config)# interface Ethernet 4/2
device(conf-if-eth-4/2)# delay-link-event 2 down
```

Link fault signaling

The Brocade SLX 9850 supports Link Fault Signaling (LFS) detection for interface types of 10G, 40G, 100G, and 40G breakout ports. It detects local and remote faults.

NOTE
LFS is not supported in 1G mode.

When the device detects a local fault, it returns a remote fault to the link partner. When the device detects a remote fault, it returns an idle state.

A port's physical link detection is independent of LFS detection. When either of these link fault signals is detected, the following behaviors occur:

- The link is declared as DOWN and the MM should display Protocol Down on the SLX-OS CLI.

- The physical link is not bought down in both of the previous cases. The peer side based on its implementation might display that the link is UP when the Brocade device displays that the link is DOWN due to a fault detection.
- The transmit (TX) packets, if any, are dropped at the MAC layer. The receive (RX) packets, if any, are dropped in the software.
- The detected signal is reported as a RASTRACE message on the line card. The same information is reported on the MM as a RASLOG. The same behavior occurs when the signal is cleared.

You can enable or disable LFS globally and on the interface level for both RX and TX directions:

- If the LFS is enabled for RX, the normal local and remote fault detection and processing described previously occur. If it is disabled for RX, local and remote fault detection are ignored.
- If the LFS is enabled for TX and a local fault occurs, a remote fault (pause frame) is generated to the remote side. If it is disabled for TX, the remote fault is not generated.

Configuring link fault signaling

By default, both TX and RX LFS are enabled. You can disable either RX or TX LFS globally or on the interface level.

Perform the following steps to configure LSF globally or on an interface.

1. From privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Globally change the LFS, if required.

   ```
   device(config)# link-fault-signaling rx off tx on
   ```

   In this example, the global LFS is disabled for the link fault RX and enabled for link fault TX.

3. Access the interface configuration mode for the port that you want to configure.

   ```
   device(config)# interface Ethernet 2/1
   ```

4. Shut down the interface.

   ```
   device(conf-if-eth-2/1)# shutdown
   ```

   The interface must be in the shutdown state before you disable or enable TX LFS.

5. Change the LFS for the interface.

   ```
   device(conf-if-eth-2/1)# link-fault-signaling rx on tx off
   ```

   In this example, the LFS for the interface is enabled for the link fault RX and disabled for the link fault TX. This configuration on the interface overrides the global configuration.

6. Enable the interface.

   ```
   device(conf-if-eth-2/1)# no shutdown
   ```

7. Verify the configuration for the interface.

   ```
   device(conf-if-eth-2/1)# do show running-config interface ethernet 2/1
   interface Ethernet 2/1
   ...
    link-fault-signaling rx on tx off
    no shutdown
   !
   ```

The following example shows the steps in the previous configuration.

```
device# configure terminal
device(config)# link-fault-signaling rx off tx on
device(config)# interface Ethernet 2/1
device(conf-if-eth-2/1)# shutdown
device(conf-if-eth-2/1)# link-fault-signaling rx on tx off
device(conf-if-eth-2/1)# no shutdown
```

Interface Ethernet ports

All Brocade device ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. In some configuration scenarios, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Displaying device interfaces

The device supports Ethernet, loopback, management, and virtual Ethernet interfaces (VEs).

Enter the **show running-config interface** command to display the interfaces and their status.

The following example displays the Ethernet interfaces on the device. They are identified by the slot number, and port number, separated by forward slashes (/). For example, the notation 1/8 indicates port 8 located in slot 1 on a chassis.

```
device# show running-config interface ethernet
interface Ethernet 1/1
 no shutdown
!
interface Ethernet 1/2
 channel-group 101 mode active type standard
 lacp timeout long
 no shutdown
!
interface Ethernet 1/3
 channel-group 101 mode active type standard
 lacp timeout short
 no shutdown
!
interface Ethernet 1/4
 shutdown
!
interface Ethernet 1/5
 shutdown
!
interface Ethernet 1/6
 shutdown
!
interface Ethernet 1/8
 channel-group 143 mode active type standard
 lacp timeout short
 no shutdown
!
interface Ethernet 1/9
 shutdown
!
```

Displaying slots and module status information

Use the **show slots** command to display information for all slots in the chassis. The information includes the slot number, and the module in the slot including its type, version, description, ID and status.

The following example shows the slot information for a Brocade SLX 9850-4 router.

```
device# show slots
Slot  Type          Description                    ID    Status
----------------------------------------------------------------
M1    MM v5         Management Module              179   ENABLED
M2                                                       VACANT
S1                                                       VACANT
S2    SFM4 v4       Switch Fabric Module           183   ENABLED
S3                                                       VACANT
S4    SFM4 v4       Switch Fabric Module           183   ENABLED
S5    SFM4 v4       Switch Fabric Module           183   ENABLED
S6                                                       VACANT
L1    LC36x100G v3  36-port 100GE card             181   ENABLED
L2    LC72X10G v5   72-port 10GE card              180   ENABLED
L3    LC36x100G v3  36-port 100GE card             181   ENABLED
L4                                                       VACANT
```

> **NOTE**
> When a slot does not include module information, it may indicate the following status:
>
> - VACANT—The slot is empty.
> - UNLATCHED—A module is inserted in the slot but it is not secured to the chassis with its captive screws. When the screws are installed, the module comes up automatically and its status changes.

Chassis and host names

A device can be identified by its IP address or by its host name and chassis name. You can customize the host name and chassis name. You can also change the default chassis IPv4 or IPv6 address.

Customizing chassis and host names

Brocade recommends that you customize the chassis name for each device. Some system logs identify the device by its chassis name; if you assign a meaningful chassis name, logs are more useful. You can also configure the host name.

To customize the chassis name and host name, perform the following steps:

1. In privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Configure the chassis name.

   ```
   device(config)# switch-attributes chassis-name SLX-market1
   ```

   A chassis name can be from 1 through 30 characters long, must begin with a letter, and can contain letters, numbers, and underscore characters.

   The default chassis name is SLX9850-# where # is the number of slots in the chassis.

3. Configure the host name.

   ```
   device(config)# switch-attributes host-name SLX-mrkt
   SLX-mrkt(config)#
   ```

   This example changes the host name to SLX-mrkt and it is displayed in the prompt.

   A host name can be from 1 through 30 characters long. It must begin with a letter, and can contain letters, numbers, and underscore characters. The default host name is SLX.

4. Exit global configuration mode.

```
SLX-mrkt(config)# exit
```

5. Verify the configuration.

```
SLX-mrkt# show running-config switch-attributes
switch-attributes chassis-name SLX-market1
switch-attributes host-name SLX-mrkt
!
```

The following configuration is an example of the previous steps.

```
device# configure terminal
device(config)# switch-attributes chassis-name SLX-market1
device(config)# switch-attributes host-name SLX-mrkt
SLX-mrkt(config)#
```

Changing the chassis IP address

If you need to move the device to a different subnet, you can change the default IPv4 or IPv6 address of the device chassis.

To change the chassis IP address, perform the following steps:

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Configure the chassis address.

```
device(config)# chassis virtual-ip 10.11.12.13/20
```

If you want to reset the address to the initial address for the chassis, use the **no** form of this command.

3. Verify the configuration.

```
device# show running-config chassis
chassis virtual-ip 10.11.12.13/20
!
```

The following configuration is an example of the previous steps.

```
device# configure terminal
device(config)# chassis virtual-ip 10.11.12.13/20
```

System clock

The operation of the device does not depend on the date and time and the Brocade device with an incorrect date and time value functions properly. However, since logging, error detection, and troubleshooting use the date and time, you should set the clock correctly.

> NOTE
> You can set the system clock if there are no NTP servers configured. Otherwise, an active NTP server, if configured, automatically updates and overrides the system clock.

Setting the clock

The Brocade device allows you to manually set the system clock. The time counter setting is retained across power cycles.

To set the clock, perform the following step:

1. In privileged EXEC mode, set the date and time.

   ```
   device# clock set 2016-07-07 12:30:45
   ```

   This example sets the time and date to 12:30:45 on July 07, 2016.

2. Enter global configuration mode.

   ```
   device# configure terminal
   ```

3. Change the time zone.

   ```
   device(config)# clock timezone America/Bogata
   ```

   This example changes the time zone to the region of America and the city of Bogata.

The following configuration is an example of the previous steps.

```
device# clock set 2016-07-07 12:30:45
device# configure terminal
device(config)# clock timezone America/Bogata
```

Management VRFs

Virtual Routing and Forwarding (VRF) is a technology that controls information flow within a network, isolating the traffic by partitioning the network into different logical VRF domains.

All management services on the Brocade device are VRF aware. The management services can select a particular VRF to reach a remote server based on a VRF. The VRFs are management (mgmt-vrf), default (default-vrf), and user defined (user-vrf).

By default, the device creates a VRF for management named mgmt-vrf and, all manageability services are accessible through this VRF. Multiple instances of IP services can be instantiated in multiple VRFs. For example, SSH can be in more than one VRF. IP services can have up to five VRF instances.

VRF reachability

The Brocade device supports the VRF reachability service. Reachability determines which VRF contains the routing information needed to reach the application servers. For example, when you configure an SSH server, you can configure the VRF information for the VRF context to resolved the SSH server route.

VRF reachability indicates the details of the VRF for servicing requests from the clients. It also indicates the clients specifying the VRF for reaching a source to ensure that the management packets are serviced or routed in a server VRF domain.

These two types of reachability services are also referred to as device-initiated and server-based services.

VRF reachability for device-initiated services

The following table lists the device-initiated services and associated commands that VRF reachability supports.

Service	VRF-related command	Additional information
Firmware download	**firmware download** [**default-config**] **ftp** | **scp** | **sftp** [**use-vrf** *vrf-name*]...	Uses TCP IPv4or IPv6. By default, mgmt-vrf is used and a user-defined VRF is optional.
Logging server	**logging syslog-server** { **ipv4** | **ipv6** *address* } [**use-vrf** *vrf-name*]	Uses TCP UDP IPv4 or IPv6.
NTP	**ntp server** *ip-address* [**use-vrf** *vrf-name*]	Uses NTP UDP IPv4.
Openflow active controller	**openflow controller** *name* **ip address** *ip-address* [**use-vrf** *vrf-name*]	Uses OpenFlow TCP IPv4. By default, mgmt-vrf is used and a user defined VRF is optional.
RADIUS	**radius-server host** *host-name* [**use-vrf** *vrf-name*]	Uses UDP IPv4 or IPv6
sFlow	[**no**] **sflow collector** *ipv4/ipv6 address port-number* [**use-vrf** *vrf-name*] [**no**] **sflow source-interface** *interface-type interface-number*	Uses sFlow UDP IPv4 or IPv6. In the case of the **sflow source-interface** command, the VRF of the specified interface is used. **NOTE** Any given interface can belong to only one VRF at any given time.
SNMP notification	**snmp-server host** *ip-address* [**use-vrf** *vrf-name*] **snmp-server v3host** *ip-address* [**use-vrf** *vrf-name*]	Uses SNMP UDP IPv4 or IPv6. By default, mgmt-vrf is used to send the SNMP notifications.
Support save	**copy support** { **ftp** | **scp** } [**use-vrf** *vrf-name*] ...	Uses TCP IPv4 or IPv6. By default, mgmt-vrf is used and a user-defined VRF is optional.
TACACS+	**tacacs-server host** *host-name* [**use-vrf** *vrf-name*]	Uses TCP IPv4 or IPv6

All these implementations use forward referencing of the VRF name in the **use-vrf** option, unless noted. At runtime when making the socket connection, the VRF ID by name must be resolved. If it does not resolve, it will result in a connection error.

VRF reachability for server-based services

The server services running on the Brocade device must listen to the requests in all the VRFs or a specified VRF and send the response back to the client in the same VRF where the request arrived. Thus, the services can come through any in-band interface bound to any VRF.

Each server-based service can have a maximum of six VRF instances including mgmt-vrf. The following table lists the server services and associated commands that VRF reachability supports.

> **NOTE**
> The SNMP server listens on all VRFs and sends the response back on the same VRF where the request arrived.
> HTTP and HTTPS are mutually exclusive on the Brocade device and both will not be enabled in different VRFs.

Service	VRF-related command	Additional information
HTTP	[**no**] **http server** [**use-vrf** *vrf-name*] [**shutdown**]	By default, the HTTP service is associated and started on mgmt-vrf and default-vrf. A user-defined VRF is optional.
SSH	[**no**] **ssh server** [**use-vrf** *vrf-name*] **shutdown**	By default, the SSH service is associated and started on mgmt-vrf and default-vrf. A user-defined VRF is optional.
Telnet	[**no**] **telnet server** [**use-vrf** *vrf-name*] **shutdown**	By default, the Telnet service is associated and started on mgmt-vrf and default-vrf. A user-defined VRF is optional.

Telnet and SSH limitations and considerations

Telnet and SSH limitations and considerations are as follows:

- By default, SSH and Telnet services are associated and started on mgmt-vrf and default-vrf.
- You cannot remove mgmt-vrf from the SSH and Telnet services.
- Telnet and SSH server can be enabled on a maximum number of 6 VRFs.
- When you use the **ssh** or **telnet server shutdown** command without a user-defined VRF, the service is shut down on mgmt-vrf only.
- Standby MM has support for mgmt-vrf. Starting SSH and Telnet service on a user-defined VRF does not have an impact on the standby MM until it becomes active.

  > **NOTE**
  > By default, SSH and Telnet service on the standby MM is disabled, use the **ssh** or **telnet server standby enable** command to enable standby access through SSH and Telnet.

- SSH and Telnet Services started on VRF context is applicable for both IPv4 and IPv6 addresses.
- A maximum of 32 user logins are allowed in the device. These sessions are a cumulative count of login sessions through SSH and Telnet across all the configured VRFs.

Network Time Protocol (NTP)

Network Time Protocol overview

Network Time Protocol (NTP) maintains uniform time across all devices in a network. The NTP commands support the configuration of an external time server to maintain synchronization between all local clocks in a network.

To keep the time in your network current, it is recommended that each device have its time synchronized with at least one external NTP server.

Date and time settings

Brocade devices maintain the current date and time inside a battery-backed real-time clock (RTC) circuit. Date and time are used for logging events. Device operation does not depend on the date and time; a device with incorrect date and time settings can function correctly. However, because the date and time are used for logging, error detection, and troubleshooting, you should set them correctly.

Time zone settings

The time zone setting has the following characteristics:

- The setting automatically adjusts for Daylight Savings Time.
- Changing the time zone on a device updates the local time zone setup and is reflected in local time calculations.
- By default, all devices are in the Greenwich Mean Time (GMT) time zone (0,0).
- System services that have already started will reflect the time zone changes only after the next reboot.
- Time zone settings persist across failover for high availability.
- Time zone settings are not affected by NTP server synchronization.

NTP server

An NTP server will provide the correct network time on your device using the Network Time Protocol (NTP). NTP can be used to synchronize the time on devices across a network.

An NTP time server is used to obtain the correct time from an external time source and adjust the local time in each connected device. When NTP server functionality is enabled, the NTP server will start listening on the NTP port for client requests and respond with the reference time. Up to five server addresses in IPv4 or IPv6 format can be configured. When you configure multiple NTP server addresses, the first obtainable address is set as the active NTP server. If there are no reachable time servers, then the local device time is the default time until a new active time server is configured.

The NTP server is stateless and will not maintain any NTP client information. Network time synchronization is guaranteed only when a common external time server is used by all devices.

NTP server authentication

You can create an authentication key for the purpose of authenticating an external Network Time Protocol (NTP) server.

The time kept on a device is a critical resource, so it is highly recommended to use the encrypted authentication mechanism. An authentication key is configured with a key identifier and secret key strings. The key is shared by the client (device) and an external NTP server by associating the key to an NTP server. NTP supports a symmetric key scheme. The scheme uses MD5 keyed hash algorithm.

Configuring NTP

After setting the date, time, and time zone on a device, the local time on a device can be synchronized with an Network Time Protocol (NTP) server.

The date, time, and time zone are all set in Privileged EXEC mode and only have to be configured once per device because the value is written to nonvolatile memory. After the basic time information is set up, an NTP server is configured to allow the local time to be synchronized across the network.

1. Set the date and time for the device.

   ```
   device# clock set 2016-08-06T12:15:00
   ```

2. Access global configuration mode.

   ```
   device# configure terminal
   ```

3. Set the time zone for the device.

   ```
   device(config)# clock timezone America/Los_Angeles
   ```

4. Access privileged EXEC mode.

   ```
   device(config)# clock timezone America/Los_Angeles
   ```

5. Display the local date, time, and time zone for the device.

   ```
   device# show clock
   2016-08-06T12:15:00 America/Los_Angeles
   ```

6. Enter global configuration mode.

   ```
   device# configure terminal
   ```

7. Synchronize the local time with an external source.

   ```
   device(config)# ntp server 192.168.10.1
   ```

8. Exit global configuration mode.

   ```
   device(config-server-19.168.10.1/mgmt-vrf)# exit
   ```

9. Exit to Privileged EXEC mode.

   ```
   device(config)# exit
   ```

10. Display the active NTP server IP address.

    ```
    device# show ntp status
    active ntp server 192.168.10.1
    ```

In the following example, the date, time and time zone are set on a device and verified. The local device is configured to synchronize the local time with an external NTP server at a specific IP address.

```
device# clock set 2016-08-06T12:15:00
device# configure terminal
device(config)# clock timezone America/Los_Angeles
device(config)# exit
device# show clock
2016-08-06T12:15:00 America/Los_Angeles
device# configure terminal
device(config)# ntp server 192.168.10.1
device(config-server-19.168.10.1/mgmt-vrf)# exit
device(config)# exit
device# show ntp status
active ntp server 192.168.10.1
```

Authenticating an NTP server

An authentication key can be created for the purpose of authenticating an external Network Time Protocol (NTP) server.

An authentication key is configured with a key identifier and secret key strings. The key is shared by the client (device) and an external NTP server by associating the key to an NTP server.

1. Enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Create an authentication key ID and key string.

    ```
    device(config)# ntp authentication-key 33 md5 check
    ```

    Up to five NTP authentication keys can be configured and each key ID must be unique.

3. Synchronize the local time with an external source, an NTP server.

    ```
    device(config)# ntp server 192.168.10.1
    ```

4. Associate the key to the NTP server.

    ```
    device(config-server-192.168.10.1/mgmt-vrf)# key 33
    ```

5. Exit to global configuration mode.

    ```
    device(config-server-192.168.10.1/mgmt-vrf)# exit
    ```

6. Exit to Privileged EXEC mode.

    ```
    device(config)# exit
    ```

In the following example, the date, time and time zone are set on a device and verified. The local device is configured to synchronize the local time with an external NTP server at a specific IP address.

```
device# configure terminal
device(config)# ntp authentication-key 33 md5 check
device(config)# ntp server 192.168.10.1
device(config-server-192.168.10.1/mgmt-vrf)# key 33
device(config-server-192.168.10.1/mgmt-vrf)# exit
device(config)# exit
```

Displaying the active NTP server

Information about the currently active NTP server can be displayed. When an NTP server has been configured, the server IP address is displayed. If an NTP server is not configured or the server is unreachable, the output displays LOCL (for local device time).

Only the local NTP server information is displayed.

NTP server status when no NTP server configured

The following example shows the local device NTP status when an NTP server is not configured:

```
device# show ntp status
active ntp server is LOCL
```

NTP server status when an NTP server is configured

The following example shows a configured NTP server:

```
device# show ntp status
active ntp server is 192.168.10.1
```

SNMP

SNMP overview

Simple Network Management Protocol (SNMP) is a set of application layer protocols for managing complex networks. Devices within a network use SNMP to send messages, called protocol data units (PDUs), to different parts of a network.

Network management using SNMP requires three components:

- SNMP manager—Typically, network management systems (NMS) that manage networks by monitoring the network parameters, and optionally, setting parameters in managed devices. The SNMP manager communicates to the devices within a network using the SNMP protocol.

- SNMP agent—Software that resides in the managed devices in the network, and collects and stores data from these devices. Each device hosts an SNMP agent. The agent receive requests from the SNMP manager and responds with the requested data. In addition, the agent can asynchronously alert the SNMP manager about events by using special PDUs called traps.

  Multiple instances of the same MIB module can support a single SNMP agent by mapping a specific key called a context name to a virtual routing and forwarding (VRF) instance created within the Brocade device.

- Management Information Base (MIB)—Hierarchical database where SNMP agents in the managed devices store the data about these devices. The MIB is is structured on the standard specified in the RFC 2578 [Structure of Management Information Version 2 (SMIv2)].

  An SNMP manager can issue read or write operations to retrieve and use the MIB objects to manage and monitor devices on the network. However, the MIB structure determines the scope of management access allowed by a device.

The SNMP server on the Brocade device supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

- SNMPv1 and SNMPv2 use community strings associated to SNMP groups. The group maps the user to MIB objects called SNMP views. The views restrict the access of the MIB OIDs .

- SNMPv3 provides additional security through authenticated users associated with groups to restrict the access of MIBs for SNMP requests through SNMP views.

Also, the device supports the configuration of trap hosts as a trap recipient to receive filtered traps based on their severity level, and optionally receive SNMP communication through a VRF.

Basic SNMP operation

Every Brocade device carries an *agent* and management information base (MIB), as shown in the next figure. The agent accesses information about a device and makes it available to an SNMP network management station.

FIGURE 1 SNMP structure
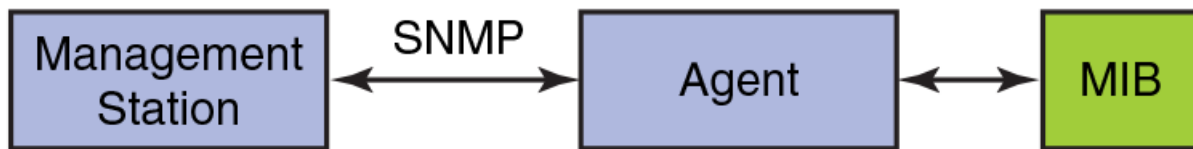
When active, the management station can "get" information or "set" information when it queries an agent. SNMP commands, such as **get**, **set**, **getnext**, and **getresponse**, are sent from the management station, and the agent replies once the value is obtained or modified as shown in the next figure. Agents use variables to report such data as the number of bytes and packets in and out of the device, or the number of broadcast messages sent and received. These variables are also known as managed objects. All managed objects are contained in the MIB.

FIGURE 2 SNMP query
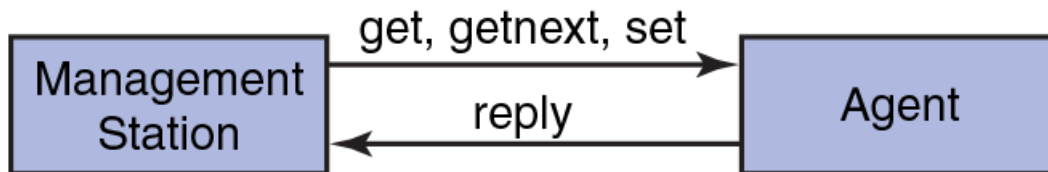
The management station can also receive *traps*, unsolicited messages from the device agent if an unusual event occurs as shown in the next figure.

FIGURE 3 SNMP trap

The agent can receive queries from one or more management stations and can send traps to up to six management stations.

SNMP community strings

SNMP versions 1 and 2 use community strings to restrict SNMP access.

You associate the community string with an SNMP group to restrict the access of MIBs for SNMPv1 and SNMPv2c requests. You can configure a total of 256 read-only and read-write community strings on the device.

The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI.

By default, you cannot perform any SNMP Set operations until you configure a read-write community string.

SNMP groups

SNMP groups map the SNMP user for SNMPv3 and the community for the SNMPv1 and SNMPv2 to SNMP views.

You can configure each group with any or all of the following views:

- Read view with read-only access
- Write view with read-write access
- Notify view to filter notifications to be encrypted and sent to target hosts

SNMP users that are mapped to a group with SNMP views use its views for access control.

SNMP users

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services. This model provides a user that is associated with security information for authentication of its generated SNMP messages.

SNMP version 3 also supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether to allow access to a managed object in a local MIB by a remote principal. You can create and associate SNMPv3 users with configured SNMP groups to use the group views for access control.

SNMP views

SNMP views are named groups of MIB objects that you can associate with groups to limit access by community strings and users for viewing and modifying the SNMP statistics and system configuration. With SNMP views, you can create or remove the access to a MIB object for inclusion or exclusion from viewing from user access.

SNMP views reference MIB objects using object names. It represents the hierarchical location of the object in the MIB tree. You associate the views with each group to restrict or allow access to the OIDs. You can create a maximum of 10 views on the device.

SNMP server hosts

On the Brocade device, the SNMP server host serves as a trap receiver to ensure that all SNMP traps sent by the device go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network.

For an SNMPv3 trap, you associate a SNMPv3 host with the SNMP users. When you specify the host, you also specify a community string for SNMPv1 and SNMPv2. The Brocade device sends all the SNMP traps to the specified hosts and includes the specified community string. Then, administrators can filter for traps from a Brocade device based on IP address or community string.

Multiple SNMP server context to VRF mapping

A single SNMP agent can support multiple instances of the same MIB module by the mapping of the context name to a virtual routing and forwarding (VRF) instance created within the device.

You map each VRF with a specific context name. The context name identifies the VRF and fetches the MIB details of the mapped VRF from the underlying modules. For example, the OSPF-MIB returns the queried OSPF-MIB object values pertaining to the default VRF (default-vrf).

For SNMPv1 and SNMPv2, the mapping of the context is with the community. This mapping is in addition to mapping of the context with the VRF. The SNMP agent supports 256 contexts to support context-to-VRF mapping.

For SNMPv3, you only need to map the context with the VRF. The SNMPv3 request PDU itself provisions for the context. Only one context is allowed for each VRF instance.

Configuring SNMPv2

SNMPv1 and SNMPv2 use community strings to restrict SNMP access. When you associate it with an SNMP group, you can restrict the access of MIBs for SNMPv1 and SNMPv2c requests.

To configure SNMPv2, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Configure the location and contact information for the SNMP server.

   ```
   device(config)# snmp-server location "Building 3 Room 214" contact "Operator 12345"
   ```

   This example changes the default location from BrocadeTcsHyd to "Building 3 Room 214" and default contact information from BrocadeCommunicationSystem to "Operator 12345".

   The double quotes allows you to enter the string with spaces.

3. Add an SNMP view to restrict or allow access to the MIB OIDs.

   ```
   device(config)# snmp-server view view2 1.3.6.1 included
   ```

   This example adds the SNMP view2 view entry with included permission to allow access for the MIB 1.3.6.1 object ID ('internet').

4. Add an SNMP group.

   ```
   device(config)# snmp-server group admin v2c write view2 notify view2
   ```

   This example adds the admin group for SNMPv2 and maps the read-write access and notify views to view2.

5. Add an SNMP community string and associate it with a group.

   ```
   device(config)# snmp-server community comm1 group admin
   ```

   This example adds the comm1 community string and associates it with the admin group to access the MIBs for SNMPv2c requests.

6. Configure the SNMP trap host associated with community string.

   ```
   device(config)# snmp-server host 10.32.147.6 comm1 version 2c severity-level Warning
   ```

   This example configures 10.32.147.6 as a trap recipient with SNMPv2c on the default target port 162 and associates the comm1 community string.

7. Enable the traps.

   ```
   device(config)# snmp-server enable trap
   ```

8. Access privileged EXEC mode.

   ```
   device(config)# exit
   ```

9. Verify the configuration.

```
device# show running-config snmp-server
snmp-server contact  "Operator 12345"
snmp-server enable trap
snmp-server location  "Building 3 Room 214"
snmp-server community comm1 group admin
snmp-server group admin v2c write view2 notify view2
snmp-server host 10.32.147.6 comm1 version 2c
 severity-level Warning
!
```

The following example shows the previous steps to configure SNMPv2.

```
device# configure terminal
device(config)# snmp-server location "Building 3 Room 214" contact "Operator 12345"
device(config)# snmp-server view view2 1.3.6.1 included
device(config)# snmp-server group admin v2c write view2 notify view2
device(config)# snmp-server community comm1 group admin
device(config)# snmp-server host 10.32.147.6 comm1 version 2c severity-level Warning
device(config)# snmp-server enable trap
```

Configuring SNMPv3

SNMPv3 uses SNMP users to restrict SNMP access. When you map an SNMP user to an SNMP group, you can restrict the access of MIBs for SNMP requests through an SNMP view.

To configure SNMPv3, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Configure the contact information for the SNMP server.

```
device(config)# snmp-server contact "Network Management group - Contact # 123-123-1234"
```

This example changes the default contact information from Field Support to "Network Management group - Contact # 123-123-1234".

The double quotes allows you to enter the string with spaces.

3. Configure the location information for the SNMP server.

```
device(config)# snmp-server location "South Room, Rack-11"
```

This example changes the default location from End User Premise to "South Room, Rack-11".

The double quotes allows you to enter the string with spaces.

4. Add an SNMP view to restrict or allow access to the MIB OIDs.

```
device(config)# snmp-server view view2 1.3.6.1 included
```

This example adds the SNMP view2 view entry with included permission to allow access for the MIB 1.3.6.1 object ID ('internet').

5. Add an SNMP group.

```
device(config)# snmp-server group group1 v3 write view2 notify view2
```

This example adds the group1 group for SNMPv3 and maps the read-write access and notify views to view2.

6. Add an SNMP user and associate it with a group.

    ```
    device(config)# snmp-server user user2 groupname group1 auth md5 auth-password private123 priv DES
    priv-password public123
    ```

    This example adds the user2 user and associates it with the group1 group to access of MIBs for SNMPv3 requests. For SNMPv3 users, the passwords for **auth-password** and **priv-password** keywords are encrypted while storing to the persistent memory or displaying it back to the user. You can configure either with a plain-text password or an encrypted password. In both cases, the **show running-config** command displays the passwords as encrypted.

7. Configure the SNMPv3 trap host associated with an SNMP user.

    ```
    device(config)# snmp-server v3host 10.26.3.166 user2 severity-level Info udp-port
    4425
    ```

    This example configures 10.26.3.166 as an SNMPv3 trap recipient host on the target port 4425 and associates the user2 user.

    The global SNMPv3 host can be associated with global SNMPv3 users only. You cannot create an SNMPv3 host in a global configuration by associating it with local SNMPv3 users.

8. Enable the traps.

    ```
    device(config)# snmp-server enable trap
    ```

9. Access privileged EXEC mode.

    ```
    device(config)# exit
    ```

10. Verify the configuration.

    ```
    device# show running-config snmp-server
    snmp-server contact "Network Management group - Contact # 123-123-1234"
    snmp-server enable trap
    snmp-server location "South Room, Rack-11"
    snmp-server group group1 v3 auth write view2 notify view2
    snmp-server user user2 groupname group1 md5 auth-password private123 priv DES priv-password public123
    snmp-server v3host 10.26.3.166 user2
     severity-level Info
     udp-port 4425
    !
    snmp-server view view2 1.3.6.1 included
    ```

The following example shows the previous steps to configure SNMPv3.

```
device# configure terminal
device(config)# snmp-server contact  "Network Management group - Contact # 123-123-1234"
device(config)# snmp-server location "South Room, Rack-11"
device(config)# snmp-server view view2 1.3.6.1 included
device(config)# snmp-server group group1 v3 auth write view2 notify view2
device(config)# snmp-server user user2 groupname group1 md5 auth-password private123 priv DES priv-password
public123
device(config)# snmp-server v3host 10.26.3.166 user2 severity-level Info udp-port 4425
device(config)# snmp-server enable trap
```

Configuring an SNMP server context to a VRF

A single SNMP agent can support multiple instances of the same MIB module by mapping the context name to a virtual routing and forwarding (VRF) instance created within the device. The SNMP context name is used to identify the VRF and fetch the MIB details of the mapped VRF from the underlying modules.

To configure an SNMP server context to a VRF for SNMPv1 or SNMPv2, perform the following steps.

NOTE
For SNMPv3, use the **snmp-server context** command only. The SNMPv3 request PDU itself has the provision for the context name as input.

ATTENTION
SNMP SET requests work only on the default VRF.

1. In privileged EXEC mode, enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Map the context with the community.

    ```
    device(config)# snmp-server community public groupname admin
    ```

3. Create a context and map it with a VRF.

    ```
    device(config)# snmp-server context mycontext vrf myvrf
    ```

4. Map the community to the context.

    ```
    device(config)# snmp-server mib community-map public context mycontext
    ```

5. Verify the configuration.

    ```
    device# show running-config snmp-server
    ...
    snmp-server community public groupname admin
    snmp-server context mycontext vrf myvrf
    ...
    snmp-server mib community-map public context mycontext
    ```

The following example shows the previous steps for the configuration.

```
device# configure terminal
device(config)# snmp-server community public groupname admin
device(config)# snmp-server context mycontext vrf myvrf
device(config)# snmp-server mib community-map public context mycontext
```

Offline SNMP ifIndex generation tool

On Brocade SLX Router, SNMP Management Information Base (MIB) uses Interface Index (ifIndex) to assign a unique identifying value to each interface.

The ifIndex is encoded per interface type and the assigned value is used if any information needs to be polled for a particular interface. The offline SNMP ifIndex generation tool which is developed based on Python, provides a means to find out the ifIndexes associated with various interfaces. This tool can run on any platform (SLX Router, Linux, or Windows) wherever the Python package is installed. The script is available on SLX Router at `/fabos/cliexec/ifindex_gen.py`. If you want to run it on a Linux or Windows platform, you may have to modify the first line in the script to point to the location of the Python binary on the platform.

Generating ifIndexes for various interfaces

ifIndex can be generated offline for various interface types such as physical interface, LAG (port-channel) interface, VE interface, loopback interface, tunnel interface and Management interface.

To generate ifIndex for a specific interface, perform the following steps.

1. Enter SLX-OS Linux shell from privileged EXEC mode.

   ```
   device# start-shell
   ```

2. Retrieve syntax to find available options to generate ifIndex.

   ```
   [admin@SLX]# /fabos/cliexec/ifindex_gen.py -h
   usage: ifindex_gen.py [-h] -i INTF_TYPE [-t LC_TYPE] [-m MODE] [-s SLOT]
                          [-p PORT] [-sp SUB_PORT] [-lp LAG_PORT]
                          [-vb VE_BRIDGE_ID] [-vi VE_INTF_ID] [-tt TUNNEL_TYPE]
                          [-ti TUNNEL_ID] [-lbi LB_INTF_ID] [-mi MGMT_INTF_ID]
                          [-d DISP_MODE]
   arguments:
   -h, --help          show this help message and exit
   -i INTF_TYPE        Interface type: [phy (physical) | lag | ve | tunnel | lb
                       (loopback) | mgmt (management)]
   -t LC_TYPE          LC type: [72x10G | 36x100G]
   -m MODE             PortGroup Mode: [40g | 100g] (required when LC type is
                       36x100G)
   -s SLOT             Slot #: [1-8]
   -p PORT             Port #: [1-72] for 72x10G, [1-60] for 36x100G LC type
   -sp SUB_PORT        Sub Port #: [1-4] for break-out ports (required when LC
                       type is 36x100G, PortGroup Mode is 40g and break-out is
                       enabled)
   -lp LAG_PORT        LAG Port #: [1-512]
   -vb VE_BRIDGE_ID    VE Bridge ID: [0-255]
   -vi VE_INTF_ID      VE Interface ID: [1-4096]
   -tt TUNNEL_TYPE     Tunnel type: [vxlan | gre | nvgre | mpls]
   -ti TUNNEL_ID       Tunnel ID: [1-1024]
   -lbi LB_INTF_ID     Loopback Interface ID: [1-255]
   -mi MGMT_INTF_ID    Management Interface ID: [1-2]
   -d DISP_MODE        Output Display Mode: [bin | dec | hex | all] (default:
                       dec)
   ```

   Note: The parameters -t, -m, -s, -p, and -sp are the sub-options specific to physical interface.

3. Generate ifIndex for a specific interface. In this example, ifIndex is generated for a physical interface.

   ```
   [admin@SLX]# /fabos/cliexec/ifindex_gen.py -i physical -t 72x10G -s 2 -p 65 -d all
   Decimal :  413171855
   Hex     :  18a0808f
   Binary  :  00011000101000001000000010001111
   ```

Configuration examples for generating ifIndexes offline

The following examples provide details on how ifIndexes can be generated for various interface types.

Physical interfaces with LC type 72x10G

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i physical -t 72x10G -s 2 -p 65 -d all
Decimal :  413171855
Hex     :  18a0808f
Binary  :  00011000101000001000000010001111
```

Physical interfaces with LC type 36x100G (100g mode)

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i phy -t 36x100G -m 100g -s 3 -p 1 -d all
Decimal :  415285249
```

```
Hex     :  18c0c001
Binary  :  00011000110000001100000000000001
```

Physical interfaces with LC type 36x100G (40g mode) non-breakout

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i phy -t 36x100G -m 40g -s 3 -p 8 -d all
Decimal :  207683777
Hex     :  0c6100c1
Binary  :  00001100011000010000000011000001
```

Physical interfaces with LC type 36x100G (40g mode) breakout

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i phy -t 36x100G -m 40g -s 3 -p 15 -sp 1 -d all
Decimal :  207741442
Hex     :  0c61e202
Binary  :  00001100011000011110001000000010
```

LAG (Port-channel) interface

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i lag -lp 1 -d all
Decimal :  671088641
Hex     :  28000001
Binary  :  00101000000000000000000000000001
```

VE interface

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i ve -vi 10 -d all
Decimal :  1207959562
Hex     :  4800000a
Binary  :  01001000000000000000000000001010
```

Tunnel interfaces

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i tunnel -tt mpls -ti 2 -d all
Decimal :  2092957698
Hex     :  7cc00002
Binary  :  01111100110000000000000000000010
```

Loopback interface

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i lb -lbi 20 -d all
Decimal :  1476395028
Hex     :  58000014
Binary  :  01011000000000000000000000010100
```

Management interface

```
[admin@SLX]# /fabos/cliexec/ifindex_gen.py -i mgmt -mi 2 -d all
Decimal :  805306370
Hex     :  30000002
Binary  :  00110000000000000000000000000010
```

LLDP

LLDP overview

The IEEE 802.1AB Link Layer Discovery Protocol (LLDP) enhances the ability of network management tools to discover and maintain accurate network topologies and simplify LAN troubleshooting in multi-vendor environments. To efficiently and effectively operate the various devices in a LAN you must ensure the correct and valid configuration of the protocols and applications that are enabled on these devices. With Layer 2 networks expanding dramatically, it is difficult for a network administrator to statically monitor and configure each device in the network.

Using LLDP, network devices such as routers and switches advertise information about themselves to other network devices and store the information they discover. Details such as device configuration, device capabilities, and device identification are advertised. LLDP defines the following:

* A common set of advertisement messages.

* A protocol for transmitting the advertisements.

* A method for storing the information contained in received advertisements.

   NOTE
   LLDP runs over the data-link layer which allows two devices running different network layer protocols to learn about each other.

LLDP information is transmitted periodically and stored for a finite period. Every time a device receives an LLDP advertisement frame, it stores the information and initializes a timer. If the timer reaches the time to live (TTL) value, the LLDP device deletes the stored information ensuring that only valid and current LLDP information is stored in network devices and is available to network management systems.

Layer 2 topology mapping

The LLDP protocol lets network management systems accurately discover and model Layer 2 network topologies. As LLDP devices transmit and receive advertisements, the devices store information they discover about their neighbors. Advertisement data such as a neighbor's management address, device type, and port identification is useful in determining what neighboring devices are in the network.

   NOTE
   The Brocade LLDP implementation supports up to two neighbors.

The higher level management tools, such as the Brocade Network Advisor, can query the LLDP information to draw Layer 2 physical topologies. The management tools can continue to query a neighboring device through the device's management address provided in the LLDP information exchange. As this process is repeated, the complete Layer 2 topology is mapped.

In LLDP the link discovery is achieved through the exchange of link-level information between two link partners. The link-level information is refreshed periodically to reflect any dynamic changes in link-level parameters. The basic format for exchanging information in LLDP is in the form of a type, length, value (TLV) field.

LLDP keeps a database for both local and remote configurations. The LLDP standard currently supports three categories of TLVs. Brocade's LLDP implementation adds a proprietary Brocade extension TLV set. The four TLV sets are described as follows:

- Basic management TLV set — This set provides information to map the Layer 2 topology and includes the following TLVs:
  - Chassis ID TLV — Provides the ID for the switch or router where the port resides. This is a mandatory TLV.
  - Port ID TLV—Provides a unique identifiable information of the port. The Port ID could be one of the following: MAC address, Network address, Interface name of the port. On the SLX-OS, the interface name of the port is provided. This is a mandatory TLV.
  - Port description TLV — Provides a description of the port in an alphanumeric format. If the LAN device supports RFC-2863, the port description TLV value equals the "ifDescr" object. This is an optional TLV.
  - System name TLV — Provides the system-assigned name in an alphanumeric format. If the LAN device supports RFC-3418, the system name TLV value equals the "sysName" object. This is an optional TLV.
  - System description TLV — Provides a description of the network entity in an alphanumeric format. This includes system name, hardware version, operating system, and supported networking software. If the LAN device supports RFC-3418, the value equals the "sysDescr" object. This is an optional TLV.
  - System capabilities TLV — Indicates the primary functions of the device and whether these functions are enabled in the device. The capabilities are indicated by two octets. The first octet indicates Other, Repeater, Bridge, WLAN AP, Router, Telephone, DOCSIS cable device, and Station, respectively. The second octet is reserved. This is an optional TLV.
  - Management address TLV — Indicates the addresses of the local switch. Remote switches can use this address to obtain information related to the local switch. This is an optional TLV.
- IEEE 802.1 organizational TLV set — This set provides information to detect mismatched settings between local and remote devices. A trap or event can be reported once a mismatch is detected. This is an optional TLV. This set includes the following TLVs:
  - Port VLANID TLV — Indicates the port VLAN ID (PVID) that is associated with an untagged or priority tagged data frame received on the VLAN port.
  - PPVLAN ID TLV — Indicates the port- and protocol-based VLAN ID (PPVID) that is associated with an untagged or priority tagged data frame received on the VLAN port. The TLV supports a "flags" field that indicates whether the port is capable of supporting port- and protocol-based VLANs (PPVLANs) and whether one or more PPVLANs are enabled. The number of PPVLAN ID TLVs in a Link Layer Discovery Protocol Data Unit (LLDPDU) corresponds to the number of the PPVLANs enabled on the port.
  - VLAN name TLV — Indicates the assigned name of any VLAN on the device. If the LAN device supports RFC-2674, the value equals the "dot1QVLANStaticName" object. The number of VLAN name TLVs in an LLDPDU corresponds to the number of VLANs enabled on the port.
  - Protocol identity TLV — Indicates the set of protocols that are accessible at the device's port. The protocol identity field in the TLV contains a number of octets after the Layer 2 address that can enable the receiving device to recognize the protocol. For example, a device that wishes to advertise the spanning tree protocol includes at least eight octets: 802.3 length (two octets), LLC addresses (two octets), 802.3 control (one octet), protocol ID (two octets), and the protocol version (one octet).
- IEEE 802.3 organizational TLV set — This is an optional TLV set. This set includes the following TLVs:
  - MAC/PHY configuration/status TLV — Indicates duplex and bit rate capabilities and the current duplex and bit rate settings of the local interface. It also indicates whether the current settings were configured through auto-negotiation or through manual configuration.
  - Power through media dependent interface (MDI) TLV — Indicates the power capabilities of the LAN device.
  - Link aggregation TLV — Indicates whether the link (associated with the port on which the LLDPDU is transmitted) can be aggregated. It also indicates whether the link is currently aggregated and provides the aggregated port identifier if the link is aggregated.
  - Maximum Ethernet frame size TLV — Indicates the maximum frame size capability of the device's MAC and PHY implementation.

LLDP configuration guidelines and restrictions

Follow these LLDP configuration guidelines and restrictions when configuring LLDP:

- Brocade's implementation of LLDP supports standard LLDP information.
- Mandatory TLVs are always advertised.
- The exchange of LLDP link-level parameters is transparent to the other Layer 2 protocols. The LLDP link-level parameters are reported by LLDP to other interested protocols.

Configuring and managing LLDP

The following sections discuss working with the Link Layer Discovery Protocol (LLDP) on Brocade devices.

Understanding the default LLDP

The following table lists the default LLDP configuration. Consider this when making changes to the defaults.

TABLE 1 Default LLDP configuration

Parameter	Default setting
LLDP global state	Enabled
LLDP receive	Enabled
LLDP transmit	Enabled
Transmission frequency of LLDP updates	30 seconds
Hold time for receiving devices before discarding	120 seconds

Disabling LLDP globally

LLDP is enabled globally by default. You can disable LLDP globally without changing any other aspect of the LLDP configuration.

To globally disable LLDP, perform the following steps:

1. From privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access LLDP configuration mode.

   ```
   device(config)# protocol lldp
   ```

3. Disable LLDP globally.

   ```
   device(config-lldp)# disable
   ```

4. Verify the configuration.

   ```
   device(config-lldp)# do show running-config protocol lldp
   protocol lldp
    system-description Brocade BR-SLX9850-4 Router
    disable
   !
   ```

The following configuration is an example of the previous steps to disable LLDP.

```
device# configure terminal
device(config)# protocol lldp
device(config-lldp)# disable
```

If required, re-enable LLDP.

```
device(config-lldp)# no disable
```

Configuring LLDP global parameters

When LLDP is enabled, the default values of its parameters are set. You can change the configuration of these parameters in LLDP configuration mode.

Specifying a system name for the Brocade device hardware

The global system name for LLDP is useful for differentiating between devices. By default, the host name from the chassis/entity management information base is used. By specifying a descriptive system name, you may find it easier to distinguish the device with LLDP. The following example changes the system name to Brocade_Alpha.

```
device(conf-lldp)# system-name Brocade_Alpha
```

Specifying an LLDP system description

The default system description depends on the device type. For example, the default description for a Brocade SLX 9850-4 router is Brocade BR-SLX9850-4 Router.

Brocade recommends that you use the operating system version for the description or use the description from the chassis/entity management information base (MIB).

Do not use special characters, such as #$!@, as part of the system name and description. The following example specifies the IT_1.6.2_LLDP_01 system description.

```
device(conf-lldp)# system-description IT_1.6.2_LLDP_01
```

Specifying a user description for LLDP

A user description for LLDP is for network administrative purposes and is not seen by neighboring devices. The following example specifies the Brocade-LLDP-installed-jan-25 description.

```
device(conf-lldp)# description Brocade-LLDP-installed-jan-25
```

Enabling and disabling the receiving and transmitting of LLDP frames

By default both transmit and receive for LLDP frames is enabled.
- The following example enables only receiving of LLDP frames.

    ```
    device(conf-lldp)# mode rx
    ```

- The following example enables only transmitting of LLDP frames.

    ```
    device(conf-lldp)# mode tx
    ```

Configuring the transmit frequency of LLDP frames

The default transmit frequency of LLDP frames is 30 seconds. You can change the frequency from 4 to 180 seconds. The following example changes the frequency to 45 seconds.

```
device(conf-lldp)# hello 45
```

Configuring the hold time for receiving devices

By default, four consecutive LLDP hello packets can be missed before removing the neighbor information. You can configure from 1 to 10 consecutive LLDP hello packets that can be missed before removing the neighbor information. The following example configures the 6 consecutive LLDP hello packets.

```
device(conf-lldp)# multiplier 6
```

Advertising the optional LLDP TLVs

By default, the port description and system name are advertised

You can advertise the rest of the optional LLDP TLVs. The following example advertises the management address, capabilities, name and description of the device, and user-configured port.

```
device(conf-lldp)# advertise optional-tlv management-address port-description system-capabilities system-
name system-description
```

Configuring the advertisement of LLDP organizationally-specific TLVs

You have the option of advertising dot1.tlv and dot3.tlv. The following example advertise dot1.tlv.

> **NOTE**
> Brocade does not recommend advertising dot1.tlv and dot3.tlv LLDPs if your network contains Converged Network Adapters (CNAs) from non-Brocade vendors. Functionality problems can occur.

```
device(conf-lldp)# advertise dot1-tlv
```

Configuring LLDP profiles

The Brocade device supports up to 64 profiles. When you configure a profile, its default parameters are from the global LLDP configuration.

1. In privileged EXEC mode, access global configuration mode.

    ```
    device# configure terminal
    ```

2. Enter LLDP configuration mode.

    ```
    device(config)# protocol lldp
    ```

3. Configure the profile name.

    ```
    device(conf-lldp)# profile UK_LLDP_IT
    ```

4. Specify a description for the profile.

    ```
    device(conf-lldp-profile-UK_LLDP_IT)#description standard_profile_by_Jane
    ```

5. Configure the transmission frequency of LLDP updates.

```
device(conf-lldp-profile-UK_LLDP_IT)# hello 10
```

6. Configure the hold time for receiving devices.

```
device(conf-lldp-profile-UK_LLDP_IT)# multiplier 2
```

7. Advertise the optional LLDP TLVs.

```
device(conf-lldp)# advertise optional-tlv system-name
```

8. Advertise the LLDP organizationally-specific TLVs.

```
device(conf-lldp-profile-UK_LLDP_IT)# advertise advertise dot1-tlv
```

   NOTE
   Brocade does not recommend advertising dot1.tlv and dot3.tlv LLDPs if your network contains CNAs from non-Brocade vendors. Functionality problems can occur.

9. Return to privileged EXEC mode.

```
device(conf-lldp-profile-UK_LLDP_IT)# end
```

10. Verify the configuration.

```
device# show running-config protocol lldp profile
 profile UK_LLDP_IT
  hello 10
  multiplier 2
  advertise dot1-tlv
  advertise option-tlv system-name
  description standard_profile_by_Jane
```

The following configuration is an example of the previous steps.

```
device# configure terminal
device(config)# protocol lldp
device(conf-lldp)# profile UK_LLDP_IT
device(conf-lldp-profile-UK_LLDP_IT)# description standard_profile_by_Jane
device(conf-lldp-profile-UK_LLDP_IT)# hello 10
device(conf-lldp-profile-UK_LLDP_IT)# multiplier 2
device(conf-lldp-profile-UK_LLDP_IT)# advertise option-tlv system-name
device(conf-lldp-profile-UK_LLDP_IT)# advertise advertise dot1-tlv
```

Configuring an LLDP profile to an interface

You can assign only one LLDP profile to an interface. If you do not use the **lldp profile** option at the interface level, the interface uses the global LLDP configuration.

To configure LLDP interface-level command options, perform the following steps.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Access the interface configuration mode.

```
device(config)# interface Ethernet 1/8
```

3.   Apply an LLDP profile to the interface.

```
device(conf-if-eth-1/8)# lldp profile network_standard
```

4.   Return to privileged EXEC mode.

```
device(conf-if-eth-1/8)# end
```

5.   Save the running-config file to the startup-config file.

```
device# copy running-config startup-config
```

The following configuration is an example of the previous steps.

```
device# configure terminal
device(config)# interface Ethernet 1/8
device(conf-if-eth-1/8)# lldp profile network_standard
```

Displaying LLDP information

The **show lldp** command allows you to display the following information:

*   LLDP status
*   LLDP neighbor information
*   LLDP statistics

Displaying LLDP status

To display the global LLDP status, use the **show lldp** command.

```
device# show lldp
LLDP Global Information
  system-name: SLX
  system-description: Brocade BR-SLX9850-4 Router
  description: Brocade-LLDP
  State:                 Enabled
  Mode:                  Receive/Transmit
  Advertise Transmitted: 30 seconds
  Hold time for advertise: 120 seconds
  Tx Delay Timer:        1 seconds
  Transmit TLVs:         Chassis ID           Port ID
                         TTL                  Port Description
                         System Name
```

To display LLDP status for an Ethernet interface, use the **show lldp interface ethernet** command.

```
device# show lldp interface ethernet 1/18
LLDP information for Eth 1/18
  State:                 Enabled
  Mode:                  Receive/Transmit
  Advertise Transmitted: 30 seconds
  Hold time for advertise: 120 seconds
  Tx Delay Timer:        1 seconds
  Transmit TLVs:         Chassis ID           Port ID
                         TTL                  Port Description
                         System Name
```

Displaying LLDP neighbor information

To display the LLDP neighbor information, use the **show lldp neighbors** command. This command allows you to display the information for all Ethernet interfaces, a specific interface, or detailed neighbor information.

The following example displays the LLDP neighbor information for all interfaces.

```
device# show lldp neighbors
Local Port  Dead Interval  Remaining Life  Remote Port ID  Remote Port Descr Chassis ID     Tx  Rx
System Name
Eth 1/18     120            102             Ethernet 2/25   Eth 2/25         768e.f807.6000  653 652  R6
Eth 1/21     120            108             Ethernet 1/21   Eth 1/21         768e.f807.6000  653 652  R6
Eth 1/40     120            110             Ethernet 1/50   Eth 1/50         768e.f807.6000  653 650  R6
Eth 1/43     120            102             Ethernet 2/51   Eth 2/51         768e.f807.6000  653 652  R6
Eth 1/50     120            102             Ethernet 2/23   Eth 2/23         768e.f807.6000  653 611  R6
```

The following example displays the LLDP neighbor information for Ethernet interface 1/18.

```
device# show lldp neighbors interface ethernet 1/18
Local Port  Dead Interval  Remaining Life  Remote Port ID  Remote Port Descr Chassis ID     Tx  Rx
System Name
Eth 1/18     120            115             Ethernet 2/25    Eth 2/25        768e.f807.6000  655 654  R6
```

The following example displays the detailed LLDP neighbor information for Ethernet interface 1/18.

```
device# show lldp neighbors interface ethernet 1/18 detail
Neighbors for Interface Eth 1/18

MANDATORY TLVs
===============
Local Interface: Eth 1/18  (Local Interface MAC: 768e.f805.5816)
Remote Interface: Ethernet 2/25 (Remote Interface MAC: 768e.f807.610d)
Dead Interval: 120 secs
Remaining Life : 118 secs
Chassis ID: 768e.f807.6000
LLDP PDU Transmitted: 656  Received: 655

OPTIONAL TLVs
==============
Port Interface Description: Eth 2/25
System Name: R6
```

Displaying LLDP statistics

To display the LLDP statistics for all interfaces or a specific interface, use the **show lldp statistics** command.

The following example displays the statistics for Ethernet interface 1/18.

```
device# show lldp statistics interface ethernet 1/18
LLDP Interface statistics for Eth 1/18
Frames transmitted: 659
Frames Aged out:    0
Frames Discarded:   0
Frames with Error:  0
Frames Recieved:    657
TLVs discarded:     0
TLVs unrecognized:  0
```

If you do not include the **interface ethernet** option, the command displays the statistics for all interfaces.

Clearing LLDP-related information

You can clear LLDP neighbor and statistic information for all interfaces or a specified interface.

To clear LLDP-related information, perform the following steps.

1. In privileged EXEC mode, clear the LLDP neighbor information.

    ```
    device# clear lldp neighbors
    ```

    This example clears the LLDP neighbor information for all interfaces.

2. Clear the LLDP statistics on an interface.

   ```
   device# clear lldp statistics interface ethernet 1/8
   ```

   This example clears the LLDP transmit and receive counters on the Ethernet interface 1/8.

3. Clear the LLDP statistics for all interfaces.

   ```
   device# clear lldp statistics
   ```

Brocade SLX 9850 Virtual Machine Access Management

Overview

The Brocade SLX 9850 allows you to access its internal host and virtual machine (VM) operating systems on the management module (MM), and the host and VM operating systems on the line cards.

The MM consist of the following operating systems:

- MMVM OS—The Linux VM on the MM (active or standby)
- TPVM OS—Third-party VM on the MM (active or standby)
- MM host OS—Hypervisor (KVM) that hosts the VM on the MM

A line card (LC) consists of the following operating systems:

- LCVM OS—Linux VM on LC
- Host OS—Hypervisor (KVM) that hosts the VM on the LC

Default credentials for the hosts

The following table provides the default user credentials to access the hosts of the operating systems on the MM and line cards. It also provides the commands to change the passwords.

Host name	Default user login	Default password	Changing the password
MMVM	root	fibranne	From the MMVM shell, the **passwd** command
	admin	password	From the SLX-OS CLI, the **username** command with RBAC rules
	user	password	
LCVM	root	fibranne	Same default passwords as in the MMVM
	admin	password	
	user	password	
MM HOST-OS	root	fibranne	From the MMVM shell, use the **hostadm** *update_passwd* command
LC HOST-OS	root	fibranne	Same default password as in the MM host OS
TPVM	root	fibranne	From the MMVM shell, use the **tpvmadm** *update_passwd* command

NOTE

LCVM users are independent of the MMVM users. Users from MMVM will not be synched to the LCVM. The following example changes the password for the admin user on the LCVM.

```
LCVM1:L2/0: >/bin/passwd admin
Enter new UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
LCVM1:L2/0: >
```

VM access

You can access the VMs and host operating systems through the SLX-OS CLI, MMVM OS shell or the serial console.

SLX-OS CLI to VM access

The SLX-OS CLI allows you to access to the following:

- MMVM shell by using the **start-shell** command
- LCVM and MM host OS through a Telnet or SSH session

When you log into the SLX-OS CLI, you can use the default admin or user credentials. Non-default users are authenticated through AAA.

The following example shows Telnet access to the SLX-OS CLI with admin credentials.

```
client# telnet 10.24.12.71
Trying 10.24.12.71...
Connected to 10.24.12.71.
Escape character is '^]'.
SLX login: admin
Password:
SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.

Welcome to the Brocade SLX Operating System Software
admin connected from 10.70.5.113 using telnet on device
device#
```

The default SLX-OS prompt is SLX#. However, throughout this guide, the device# is used as the prompt.

To exit the session, enter **exit**.

Accessing the Linux shell from the SLX-OS CLI

With the admin role, you can access the MMVM Linux shell from the SLX-OS CLI by using the **start-shell** command.

NOTE

The **start-shell** command is only visible in the CLI when you are configured as a user with the admin role.
Inside the MMVM Linux shell, you will have root level privilege and can run all supported Linux commands.

Perform the following steps to access the shell.

1. From privileged EXEC mode, access the MMVM Linux shell.

   ```
   device# start-shell
   Entering Linux shell for the user: admUser

   We trust you have received the usual lecture from the local System
   Administrator. It usually boils down to these three things:

       #1) Respect the privacy of others.
       #2) Think before you type.
       #3) With great power comes great responsibility.

   [admUser@SLX]#
   ```

   You can also run this command from global configuration mode by entering the **do start-shell** command.

   > **NOTE**
   > The idle timeout of the MMVM Linux shell session is five minutes. If the MMVM Linux shell is idle for more than five minutes, you are automatically logged out of Linux shell and returned to SLX-OS CLI.

2. Exit the shell and return to the SLX-OS CLI.

   ```
   [admUser@SLX]# exit
   ```

   Upon exiting, the following message appears and you return to the SLX-OS CLI prompt.

   ```
   exit

   Exited from Linux shell

   device#
   ```

Accessing LCVM from the SLX-OS CLI

You can access the LCVM on a line card from the SLX-OS CLI from a Telnet or an SSH session. In the LCVM shell, you can execute any LC command.
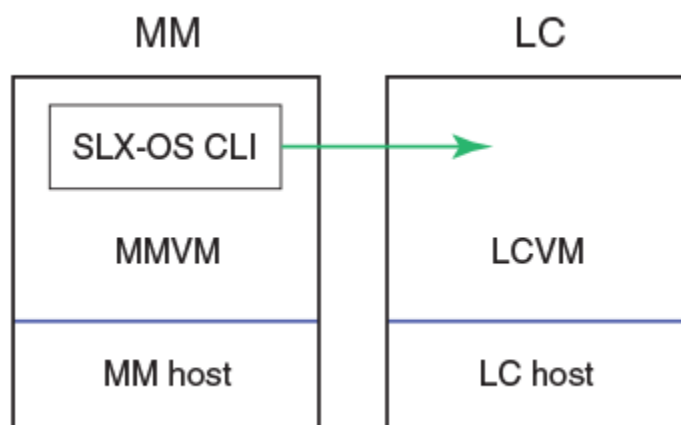
In the following example, a Telnet session is used to access the LCVM on line card 2.

```
device# telnet lc2_vm1 vrf mgmt-vrf
/fabos/cliexec/vrfcmd 0 /usr/bin/telnet lc2_vm1 23
```

```
Trying 127.2.1.4...
Connected to lc2_vm1.
Escape character is '^]'.
Ubuntu 14.04 LTS
LCVM1 login: admin
Password:
LCVM1:L2/0: >
LCVM1:L2/0: >exit
logout
Connection closed by foreign host.
device#
```

The default LCVM OS shell prompt for LC2 VM1 is LCVM1:L2/0: >.

In the following example, an SSH session is used to access the LCVM on line card 2.

```
device# ssh lc2_vm1 vrf mgmt-vrf -l admin
The authenticity of host 'lc2_vm1 (127.2.1.4)' can't be established.
ECDSA key fingerprint is d7:d9:56:82:28:5c:58:71:90:5a:e2:ec:d7:f7:db:18.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lc2_vm1' (ECDSA) to the list of known hosts.
admin@lc2_vm1's password:
LCVM1:L2/0: >
LCVM1:L2/0: >exit
logout
Connection to lc2_vm1 closed.
device#
```

In the previous examples, use **exit** to terminate the session and return back to the parent shell.

Accessing an MM host-OS shell from the SLX-OS CLI

You can access the MM host-OS shell from the SLX-OS CLI from a Telnet or an SSH session.
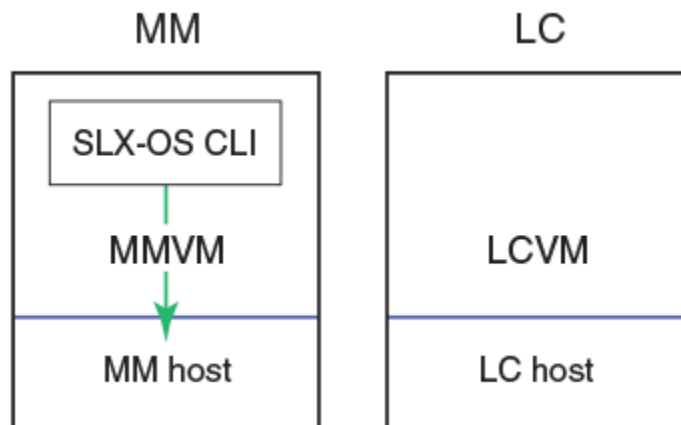
In the following example, a Telnet session is used to access the MM2 host-OS shell.

```
device# telnet mm1_kvm vrf mgmt-vrf
/fabos/cliexec/vrfcmd 0 /usr/bin/telnet mm1_kvm 23
Trying 127.2.0.1...
Connected to mm1_kvm.
Escape character is '^]'.
Ubuntu 14.04 LTS
HOST login: root
Password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.14.17 x86_64)
 * Documentation:  https://help.ubuntu.com/
   System information as of Wed Aug  3 12:41:01 PDT 2016
```

```
    System load: 0.08              Memory usage: 0%   Processes:       164
    Usage of /:  14.6% of 9.72GB   Swap usage:   0%   Users logged in: 0
    => There is 1 zombie process.
    Graph this data and manage this system at:
       https://landscape.canonical.com/
  The programs included with the Ubuntu system are free software;
  the exact distribution terms for each program are described in the
  individual files in /usr/share/doc/*/copyright.

  Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
  applicable law.
  root@HOST:~#
```

In the following example, an SSH session is used to access the MM2 host-OS shell.

```
  device# ssh mm1_kvm vrf mgmt-vrf -l root
  The authenticity of host 'mm1_kvm (127.2.0.1)' can't be established.
  ECDSA key fingerprint is d5:ba:cc:d4:57:03:e4:b2:6f:ca:d2:dd:4c:40:5d:60.
  Are you sure you want to continue connecting (yes/no)? yes
  Warning: Permanently added 'mm1_kvm' (ECDSA) to the list of known hosts.
  root@mm1_kvm's password:
  Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.14.17 x86_64)
   * Documentation:  https://help.ubuntu.com/
    System information as of Thu Aug  4 10:41:47 PDT 2016
    System load:  0.81              Users logged in:      0
    Usage of /:   14.7% of 9.72GB   IP address for eth1:   127.2.0.1
    Memory usage: 54%               IP address for br0:    10.24.12.77
    Swap usage:   0%                IP address for virbr0: 192.168.122.1
    Processes:    164
    Graph this data and manage this system at:
       https://landscape.canonical.com/
  Last login: Thu Aug  4 10:41:47 2016 from cp0_vm1
  root@HOST:~#
```

In the previous examples, use **exit** to terminate the session and return back to the parent shell.

MMVM shell to another MM or LC VM access

The MMVM Linux shell allows you access another MM or the LC VM by using their host names and credentials through a Telnet or SSH session. The following table lists the hosts and addresses, and the session type in which the MMVM shell can access them through the Ethernet 1 backplane.

Host name	Description	IP address	MMVM accessibility
mm1_vm	MM1 MMVM	127.2.1.1	Telnet or SSH
mm2_vm	MM2 MMVM	127.2.1.2	Telnet or SSH
lc1_vm1 through lc8_vm1	LC1 through LC8 LCVM	127.2.1.3 through 127.2.1.10	Telnet or SSH
mm1_kvm	MM1 host OS	127.2.0.1	SSH only
mm2_kvm	MM2 host OS	127.2.0.2	SSH only
lc1_kvm through lc8_kvm	LC1 through LC8 host OS	127.2.0.3 through 127.2.0.10	SSH only

The following example shows Telnet access of LC1 VM from the MMVM shell.
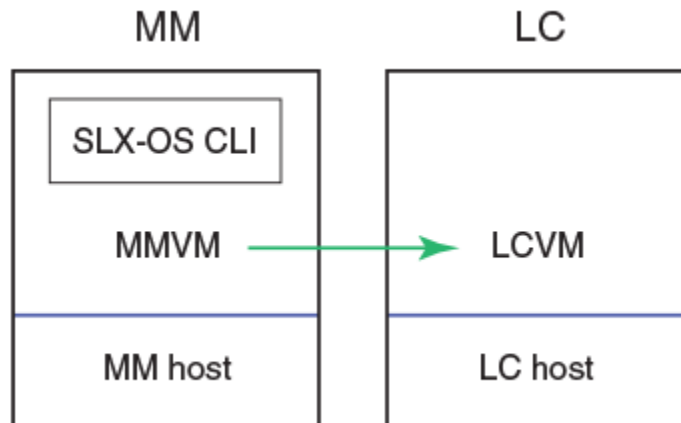
```
[admin@SLX]# telnet lc1_vm1
Trying 127.2.1.3...
Connected to lc1_vm1.
Escape character is '^]'.
Ubuntu 14.04 LTS

LCVM1 login: admin
Password:
LCVM1:L1/0: >
```

In the following example, SSH is used to access LC1 VM from the MMVM shell.

```
[admin@SLX]# ssh lc1_vm1 -l admin
admin@lc1_vm1's password:
LCVM1:L1/0: >
LCVM1:L1/0: >
```

Serial console to VM access

The serial console on the MM allows access to the following:

- MMVM OS
- MM host OS
- TPVM
- LCVM
- LC host OS

On the MM, initial authentication occurs at the SLX-OS CLI prompt.

```
SLX login: admin
Password:

SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.

Welcome to the Brocade SLX Operating System Software
admin connected from 127.0.0.1 using console on SLX
device#
```

Each OS requires its own credentials.

Using the serial console to access the MMVM, MM host OS, or TPVM

Within the MM, shortcut keys allow access to the MMVM, MM host OS, or TPVM operating systems.

Operating system	Shortcut keys
Host OS	Ctrl-Y1
MMVM OS	Ctrl-Y2
TPVM	Ctrl-Y3
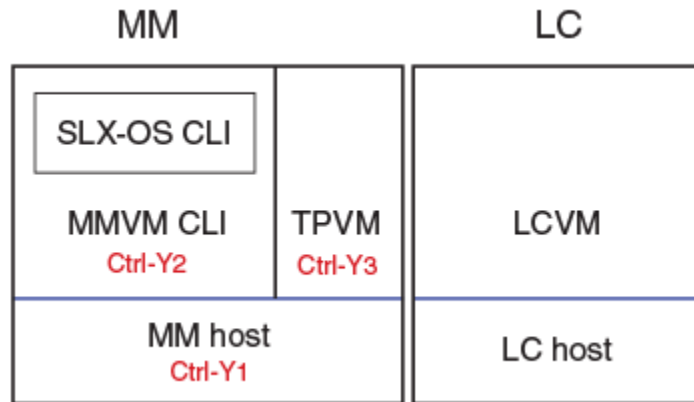
Serial console to host OS access example

```
device# Ctrl-Y1
Ubuntu 14.04 LTS HOST ttyS0

HOST login: root
Password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.14.17 x86_64)
…
…
root@HOST:~#
```

Serial console to MMVM shell access example

```
device# Ctrl-Y2
[admin@SLX]#
```

Serial console to TPVM access example

From the MMVM shell, start TPVM.

```
[admin@SLX]# tpvmadm install
Installation starts. To check the status, run 'tpvmadm show'
[admin@SLX]# tpvmadm show
TPVM is installed but not running, and AutoStart is disabled on this host.
[admin@SLX]# tpvmadm start
start succeeds
[admin@SLX]#
```

Authenticate and access the TPVM shell prompt.

```
[admin@SLX]# Ctrl-Y3
TPVM login: root
Password:
```

```
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)
…..
root@TPVM:~#
root@TPVM:~# exit

Ubuntu 14.04.3 LTS TPVM ttyS0

TPVM login:
```

To return to the MMVM CLI shell prompt, enter Ctrl-Y2.

Using the serial console to access the MMVM, MM host OS, or TPVM

From the SLX-OS CLI on the serial console, you use a remote console to access the LCVM and LC host OS. The following figure shows access to LCVM with the **rconsole 2** command.
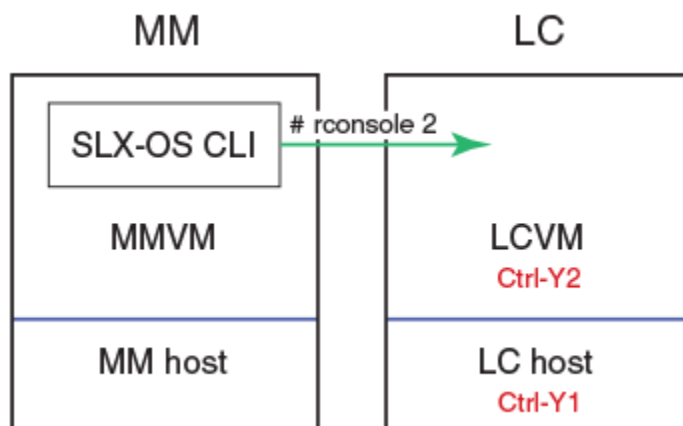
After you access the LCVM, you can use the Ctrl-Y1 shortcut key to access the LC host OS.

The following shortcut keys are available on the line card to move between the LCVM and LC host OS.

Operating system	Shortcut keys
LC host OS	Ctrl-Y1
LCVM	Ctrl-Y2

The following example shows console access to LCVM on line card 2.

```
device# # rconsole 2
Connecting to Linecard L2
Connecting to /dev/ttyS1, speed 9600
 Escape character: Ctrl-\ (ASCII 28, FS): enabled
Type the escape character followed by C to get back,
or followed by ? to see other options.
---------------------------------------------------

Ubuntu 14.04 LTS LCVM1 ttyS0

LCVM1 login: admin
Password:
LCVM1:L2/0: >
```

To access the LC host OS, use Ctrl-Y1.

To return to the LCVM CLI shell prompt, enter Ctrl-Y2.

TVPM address assignment

TPVM has an Ethernet 0 and 1 interfaces. Ethernet 0 (eth0) is enabled and its IP address is configured using DHCP. On the MMVM, the **tpvmadm show_ipaddr** command configures the address.

If DHCP is not available, you can use the serial console to log in as root, and use standard Ubuntu (14.04 LTS) commands to configure the IP address for eth0 interface.

Eth1 is connected to the Insight interface and allows packet monitoring.

Management Module High Availability

Management module HA overview

High availability (HA) is supported on management modules (MMs).

> **NOTE**
> High availability is not supported for In-Service Software Upgrade (ISSU).

The following figure shows the management module failover (MMFO) process on a device with two management modules; MM1 and MM2.

Initially, the SLX-OS VM on MM1 is active. On failover, the SLX-OS VM on the standby module (MM2) becomes active and the SLX-OS VM on MM1 reboots as standby.

FIGURE 4 Management module failover
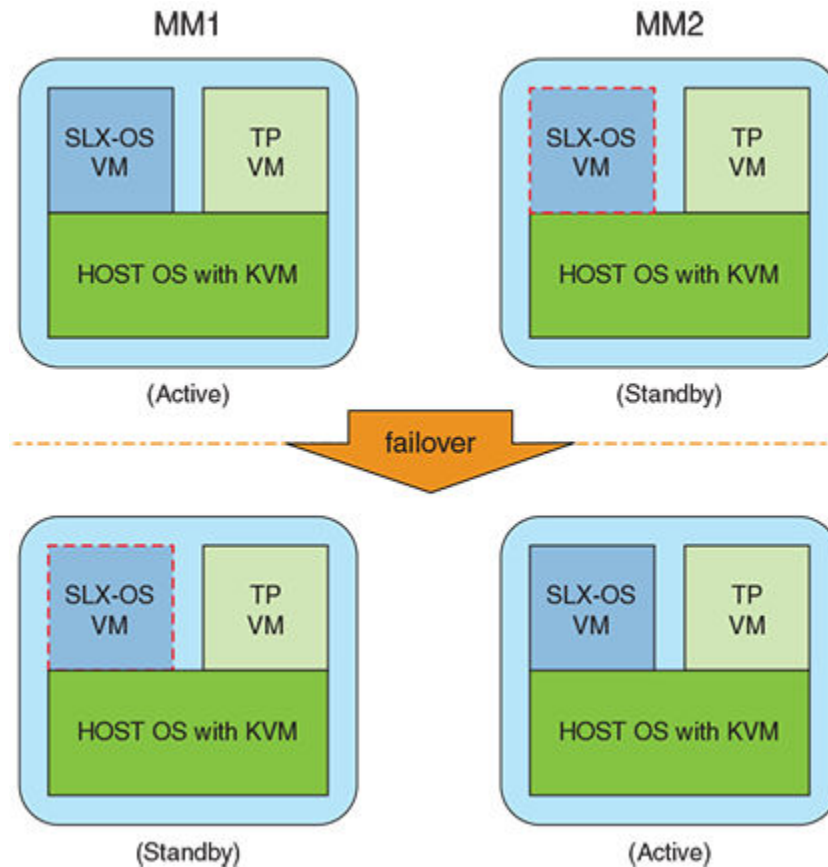
A controlled MMFO occurs when you execute a command on the active MM to instigate the failover. A MMFO that occurs for any other reason is uncontrolled.

Normally, when MMFO occurs, active and standby MMs are synchronized. However, it is possible that for some reason the active and standby MMs are not synchronized when MMFO occurs. MM behavior upon failover is dependant on whether or not the active and standby MMs are synchronized and is shown in separate columns in the following tables that describe controlled and uncontrolled MMFOs. In these tables, the term "cold recovery" indicates that line cards are rebooted and traffic is disrupted on MMFO and the term "warm recovery" describes a hitless failover in which line cards are not rebooted and traffic is not disrupted on MMFO. When HA failover occurs, there maybe a ping loss of 4 - 5 milliseconds.

A variety of HA commands are available on the device in privileged EXEC mode:

- The **show ha** command displays the management module status.
- The **ha failover** command forces the active management module to fail over. The standby management module will take over as the active management module. This command is only available in a modular chassis system.
- The **reload system** command reboots the entire chassis. This command is supported only on the active management module. This command is not supported on the standby management module. Both management modules must be in sync for the HA reboot operation to succeed.
- The **ha sync start** command enables HA state synchronization after an **ha sync stop** command has been invoked.

The following table shows commands that trigger controlled MMFO. It shows the impact of each command on active and standby MMs when the active and standby modules are synchronized and when they are not synchronized.

The **ha failover** command is the recommended method of executing a controlled MMFO.

TABLE 2 MM behavior in a controlled MMFO

Command	HA in sync		HA not in sync	
	Active MM	Standby MM	Active MM	Standby MM
reload	SLX-OS VM reboot itself can cause a cold recovery on the standby SLX-OS VM. Running-config will be used in new active.	SLX-OS VM self-reboot.	Prompt user to do 'reload system'.	SLX-OS VM self-reboot.
reload standby	Reboots standby SLX-OS VM.	SLX-OS VM self-reboot.	Reboots standby SLX-OS VM.	SLX-OS VM self-reboot.
reload system	Reboot chassis (including SLX-OS, TPVM and Host OS on both MMs) and retain mastership. Startup-config will be used.	Not allowed.	Reboot chassis (including SLX-OS, TPVM and Host OS on both MMs) and retain mastership. Startup-config will be used.	Not allowed.
ha failover	Reboots active SLX-OS VM and causes a warm recovery on the standby SLX-OS VM.	Not allowed.	Not allowed.	Not allowed.

The following table describes events that cause uncontrolled MMFO. It shows the impact of each event on active and standby MMs when the active and standby modules are synchronized and when they are not synchronized.

TABLE 3 MM behavior in an uncontrolled MMFO

Event	HA in sync		HA not in sync	
	Active MM	Standby MM	Active MM	Standby MM
Panic on active MM SLX-OS	Reboots active SLX-OS VM and causes a warm recovery on standby SLX-OS VM.	Not applicable.	Reboot Active SLX-OS VM and retain mastership. Startup- config will be used. Standby SLX-OS VM is rebooted.	Not applicable.

TABLE 3 MM behavior in an uncontrolled MMFO (continued)

Event	HA in sync		HA not in sync	
Pull out active MM	Causes a warm recovery on standby SLX-OS VM.	Not applicable.	Reboot standby SLX-OS VM and it will boot up in active. Startup- config will be used.	Not applicable.

Traffic forwarding features support for HA

Traffic forwarding features that support high availability (HA) are not disrupted when a switchover occurs between the active and standby management modules (MMs). Features that do not support HA may be disrupted.

The following traffic forwarding features support high availability (HA) and traffic is not disrupted upon MM switchover.

TABLE 4 Traffic forwarding features that support HA

Traffic forwarding feature	Note
802.1x	
BGP and BGP(v6) - NSF/GR	
CAM profiles	Supports HA without any traffic loss.
CFM/802.1ag	
Insight Architecture	Requires two guest virtual machines and applications that run independently on the active and standby MMs.
IPv4 and IPv6 forwarding	Requires routing protocols that support HA such as OSPF, BGP, ISIS GR, or NSR.
IP over MPLS	Requires: • Routing protocols that support HA • LDP graceful restart (GR) enabled
ISIS and ISIS(v6) - NSR	
L2 ACL, IPv4 ACL, IPv6 ACL, IPv4 rACL, IPv6 rACL, IPv4 PBR, IPv6 PBR, VXLAN visibility	Supports HA without any traffic loss.
LACP	
Layer-2 forwarding	
Layer-2 multicast	
MPLS transit LSR	LDP GR must be enabled.
OSPFv2 - NSF/GR , NSR	
OSPFv3 - NSR	
QoS-mutation features (CoS, DSCP, MPLS EXP)	Supports HA without any traffic loss.
SFLOW and rate-limiting features (port-based, BUM, and ACL-based)	Supports HA without any traffic loss.
UDLD	
VLAN and bridge-domain local switching, VLAN MAC port security	Supports HA without any traffic loss.
VPLS, VLL (using LDP tunnel as the transport mechanism)	LDP graceful restart (GR) must be enabled. **NOTE** VPLS or VLL configurations that use RSVP LSPs do not support HA and traffic disruption may occur.
VRF-LITE	
xSTP	

The following traffic forwarding features do not support HA; active management module failover may result in traffic disruption. However, normal operation resumes once the switchover is complete and when sessions or protocols are re-established.

TABLE 5 Traffic forwarding features that do not support HA

Traffic forwarding feature	Note
All multicast features	After MM failover, traffic disruptions may occur until the IP routing table is rebuilt, IGMP or PIM states are rebuilt, and the hardware is reprogrammed.
VRRP/E	Traffic disruption may occur in the scenario where: 1. The active MM is master 2. The remote node becomes master 3. The standby MM comes up and becomes master again There is no traffic disruption when the active MM is backup.
GRE (IPv4 and IPv6)	Traffic disruption occurs when the tunnel goes down and comes back up.
Openflow	During MM failover, flows are deleted. Flows are reinstalled after the switchover is complete.
MCT (for Layer 2 and Layer 3 traffic)	
Applications statistics	Counters are reset to zero upon MM failover.
VPLS, VLL (using RSVP LSPs)	

Insight Interface

Third-party virtual machine

The third-party virtual machine (TPVM) is one of the key components of the insight interface feature.

A TPVM resides on an SLX-OS device. It connects to each of the SLX-OS management modules (MM1 and MM") through the insight interface.

TPVM deployment in the management module

The TPVM is a dedicated VM on the device that is separate from the management module (MM) VM running the operating system.

Consider the following interactions of TPVM and the MMVM when configuring a TPVM:

- TPVM co-exists with the MMVM.
- The hosts common management bridge (br0) provides access to both the TPVM and the MMVM.
- The TPVM management IP address is DHCP assigned by default.
- The Ethernet controller associated with the TPVM insight interface port is directly injected to the TPVM using Intel Virtualization Technology for Directed I/O (VT-d).
- Virtual disks are files in the host Linux file system but appear as real hard disks to the VMs.
- Both the MM1 and the MM2 can host a TPVM independently. There is no redundancy relationship between them.
- TPVM is not supported in line cards (LC).
- No high availability support is provided for the TPVM and applications running on the TPVM.

The TPVM is not a part of the SLX-OS, although it is physically running on an MM CPU. TPVMs running on MM1 and MM2, as well as their applications, run as separate independent VMs that share CPU, RAM, hard drive, and management resources with the SLX-OS VM.

TPVM supported third-party applications, packages, and hardware

TPVM supports the following.

Brocade SLX-OS provides support for built-in applications (i.e. third-party packages shipped with the SLX-OS) that are listed in the Brocade SLX-OS Management Configuration Guide.

- Brocade is committed to providing limited support for the interoperability of these applications with Brocade application interfaces.
- Brocade does not provide support for the application configuration, functionality, and/or deficiencies.
- Brocade does not provide any support for applications not listed in the Brocade SLX-OS Management Configuration Guide.

Supported third-party applications

This feature supports the following third-party applications:

- Packet capture applications
- RESTful support to access SLX-OS

Supported third-party packages

This feature supports the following third-party packages:

- Packages installed by default on the TPVM
- RESTful application: Chrome browser – GUI RESTful access
- RESTful application: cURL – Command line RESTful access
- Tcpdump – Command line packet capture utility
- Tshark – Command line packet capture utility
- Wireshark – GUI packet capture utility

Hardware support

The TPVM has 8 GB RAM and a 256 GB solid-state disk (SSD), which limits amount of data captured through packet capture applications. To overcome this limitation, Brocade provides support for the NFS mount of an external drive.

TPVM installation

TPVM is installed and run from the management module VM (MMVM)

TPVM is released as a Debian package. You must copy this package to the `/tftpboot` directory of the MMVM. Once the TPVM Debian package is available in `/tftpboot`, use the **tpvmadm** command, available at the MMVM Linux command prompt, to install, start, and stop the TPVM.

See the sections Common TPVM commands on page 80, Assigning a static IP address on the TPVM Linux OS on page 90 and TPVM troubleshooting commands on page 87 and your Linux manual for more information about Linux TPVM commands.

Common TPVM commands

The **tpvmadm** command, run on the management module VM (MMVM). is used to control all aspects of the TPVM.

General commands

Recall that the host's common management bridge (br0) provides management access to both the TPVM and MMVM.

Move into the `tftpboot` directory.

```
device# cd /tftpboot
```

Download the package from a TFTP server.

```
device# tftp -g -r TPVM-install-package-name 10.10.10.1
```

Install the TPVM.

```
device# tpvmadm install mm1
```

Start the TPVM.

```
device# tpvmadm start mm1
```

Stop the TPVM.

```
device# tpvmadm stop mm1
```

Help commands

Get help for a command.

```
device# tpvmadm help
Usage: tpvmadm commands [ mm1 | mm2 ]
commands:
    install         install TPVM
    uninstall       uninstall TPVM
    start           start TPVM
    stop            stop TPVM
    add_disk        add a new disk to TPVM
    remove_disk     remove the disk from TPVM
    show_disk       show the disk information on TPVM
    enable_on_boot  enable to start TPVM at boot
    disable_on_boot disable to start TPVM at boot
    show            show status of TPVM
    help            show command usage.
```

Get help with the installation.

```
device# tpvmadm install help
    tpvmadm install [ mm1 | mm2 ] -- install TPVM on MM1 or MM2
    tpvmadm install              -- install TPVM on this host
==============================================================================================

device# tpvmadm install
```

> **NOTE**
> After the command is executed and the installation starts, you can check the status by running the **tpvmadm show** command.

Get help with uninstalling.

```
device# tpvmadm uninstall help
    tpvmadm uninstall [ -f ] [ mm1 | mm2 ] -- uninstall TPVM on MM1 or MM2
    tpvmadm uninstall [ -f ]               -- uninstall TPVM on this host
    -f: clear installation or uninstallation error(s) then try to uninstall (forcefully)
        example: tpvmadm uninstall -f
==============================================================================================

device# tpvmadm uninstall
```

Get help with starting the devices.

```
device# tpvmadm start help
    tpvmadm start [ mm1 | mm2 ] -- start TPVM on MM1 or MM2

    tpvmadm start              -- start TPVM on this host
==============================================================================================

device# tpvmadm start
```

Get help with stopping the devices.

```
device# tpvmadm stop help
    tpvmadm stop [ mm1 | mm2 ] -- stop TPVM on MM1 or MM2
    tpvmadm stop              -- stop TPVM on this host
==============================================================================================
```

```
device# tpvmadm stop mm1
```

Get help with enabling a disk.

```
device# tpvmadm enable_on_boot help
    tpvmadm enable_on_boot [ mm1 | mm2 ] -- enable to start TPVM at boot on MM1 or MM2
    tpvmadm enable_on_boot             -- enable to start TPVM at boot on this host
================================================================================================

device# tpvmadm disable_on_boot
```

Get help with disabling a disk.

```
device# tpvmadm disable_on_boot help
    tpvmadm disable_on_boot [ mm1 | mm2 ] -- disable to start TPVM at boot on MM1 or MM2
    tpvmadm disable_on_boot             -- disable to start TPVM at boot on this host
================================================================================================

device# tpvmadm disable_on_boot
```

Get help with updating a password.

```
device# tpvmadm update_passwd help
    tpvmadm update_passwd [ mm1 | mm2 ]  -- update the TPVM root password on MM1 or MM2
    tpvmadm update_passwd             -- update the TPVM root password on this host
================================================================================================

device# tpvmadm update_passwd mm2
root password: <enter a new root password here..>
re-enter root password: <enter the root password again..>
```

Show commands

Get help with the **show** command.

```
device# tpvmadm show help
    tpvmadm show [ -c error ] [ mm1 | mm2 ] -- show status or errors of TPVM on MM1 or MM2
    tpvmadm show [ -c error ]             -- show status or errors of TPVM on this host
    -c error: clear the runtime error when the 'sub-command' ran
    example: tpvmadm show -c remove_disk
```

Get help with displaying disk information.

```
device# tpvmadm show_disk help
    tpvmadm show_disk <disk_name|all> [ mm1 | mm2 ] -- show the disk information on MM1 or MM2
    tpvmadm show_disk <disk_name|all>             -- show the disk information on this host
    disk_name: the disk name whose information is shown.
               the name must be 'vd[b-x]' or 'all'
               example: vdb
               When 'all' is passed, the information about all disks is shown.
================================================================================================

device# tpvmadm show_disk all
disk: vdb
Capacity: 70.00 GiB
Allocation: 196.00 KiB

disk: vdc
Capacity: 30.00 GiB
Allocation: 196.00 KiB

total:
Capacity: 100.00 GiB
Allocation: 100.00 GiB
Available: 0.00 B
device# tpvmadm show_disk vdb
disk: vdb
Capacity: 70.00 GiB
```

```
Allocation: 196.00 KiB

total:
Capacity: 100.00 GiB
Allocation: 100.00 GiB
Available: 0.00 B
```

Get help with displaying the IP address.

```
device# tpvmadm show_ipaddr help
    tpvmadm show_ipaddr [ mm1 | mm2 ]  -- show TPVM IP address(es) on MM1 or MM2
    tpvmadm show_ipaddr                -- show TPVM IP address(es) on this host
=============================================================================================

device# tpvmadm show_ipaddr
IPv4:
eth0 10.20.99.162
IPv6:
eth0 2620:100:0:f803:629c:9fff:fe01:ff10
eth0 fec0:60:69bc:87:629c:9fff:fe01:ff10
eth0 fe80::629c:9fff:fe01:ff10
eth1 fe80::7:d0ff:fe02:100
```

Display the current status of tMM1.

```
device# tpvmadm show mm1
```

Show the status of the TPVM.

```
device# tpvmadm show
TPVM is running, and AutoStart is disabled on this host.
```

Example of using the **show** command for troubleshooting. In this instance the problem is a runtime error that resulted from a failure to add a disk.

```
device# tpvmadm add_disk auto 10g
add_disk failed
=============================================================================================

device# tpvmadm show
TPVM had runtime error(s) -- these error(s) seem not fatal, and the operation(s) could be
retryable add_disk: virsh vol-create-as failed. error detail: error: Failed to create vol
vde error: operation failed: the number of volumes goes beyond the maximum

TPVM is running, and AutoStart is disabled on this host.
=============================================================================================

device# tpvmadm show -c add_disk
TPVM is running, and AutoStart is disabled on this host.
```

> **NOTE**
> This particular runtime error is removed automatically when the command succeeds.

Disk management

Add a disk. This command picks the next logical disk name automatically, followed by a verification command.

```
device# tpvmadm add_disk auto 35g
add_disk succeeds
=============================================================================================

device# tpvmadm show_disk all mm2
disk: vda
Capacity: 40.00 GiB
Allocation: 136.01 MiB

disk: vdb
```

```
Capacity: 25.00 GiB
Allocation: 183.01 MiB

disk: vdc
Capacity: 35.00 GiB
Allocation: 196.00 KiB

total:
Capacity: 100.00 GiB
Allocation: 100.00 GiB
Available: 0.00 B
```

Get help with adding a disk.

```
device# tpvmadm add_disk help
    tpvmadm add_disk <disk_name|auto> <disk_size> [ mm1 | mm2 ] -- add a new disk to MM1 or MM2
    tpvmadm add_disk <disk_name|auto> <disk_size>              -- add a new disk to this host
    disk_name: the disk name added to TPVM.
               the name must be 'vd[b-x]' or 'auto'
               example: vdb
               When 'auto' is passed, the system assigns the next disk automatically.
               Otherwize, the disk name must be the next disk.
               For example, if the last disk added to the system is 'vdb', the disk name must be 'vdc'.
    disk_size: any positive number.
               Also the following suffix can be added to the end.
               b or B for bytes
               k or K for kilo bytes
               m or M for mega bytes
               g or G for giga bytes
               example: 10, 1024b, 1k, 10m, 100g
               When no suffix is used, the size is taken as giga bytes. For example, '5' means '5g'.
================================================================================================

device# tpvmadm add_disk auto 10g
or
device# tpvmadm add_disk vdd 512m
```

> **NOTE**
> The maximum number of disks is currently three. If the number of the allocated disks exceeds three, the **add_disk** command fails.

Get help with removing a disk.

```
device# tpvmadm remove_disk
    tpvmadm remove_disk <disk_name|auto> [ mm1 | mm2 ] -- remove the last disk from MM1 or MM2
    tpvmadm remove_disk <disk_name|auto>               -- remove the last disk from this host
    disk_name: the disk name removed from TPVM.
               the name must be 'vd[b-x]' or 'auto'
               example: vdb
               When 'auto' is passed, the system removes the latest disk automatically.
               Otherwise, the disk name must be the last disk added to the system.
               For example, if the last disk added to the system is 'vdx', the disk removed from
               the system must be this disk, 'vdx'.
================================================================================================

device# tpvmadm remove_disk auto
'umount' is needed before this disk is removed. Continue? [y/n]: y
or
device# tpvmadm remove_disk vdc
'umount' is needed before this disk is removed. Continue? [y/n]: y
```

> **NOTE**
> You must unmount the disk before it is removed from the system. Otherwise, the next disk you add is labeled differently and/or wrongly. If the system falls because of a mounting error, you must reboot TPVM to recover.

TPVM access and limitations

TPVM will be provided with eth0 (management) and eth1 (insight) interfaces.

The eth0 interface's bridged networking, allows it to connect to the outside network through the host physical interface and make it appear as normal host to the rest of the network.

The eth1 interface is primarily used for delivering data packets from the line card front panel ports to the TPVM.

SSH/Telnet accesses the TPVM uses the eth0 interface's IP address configured on the TPVM. You cannot Telnet or SSH into MM2 TPVM through an active MM1 management interface and vice versa.

Access and OS

The Brocade SLX 9850 accesses its internal host and VM operating systems on the MM, and the host and VM operating systems on the LCs.

The MMs run on the following operating systems:

1. Management module VM (MMVM) OS — The Linux VM on the MM (active or standby).

2. MM host OS — Hypervisor kernel-based VM (KVM) that hosts the VM on the MM.

3. TPVM OS — TPVM on the MM (active or standby).

A LC runs on the following operating systems:

1. LCVM OS — Linux VM on the LC.

2. Host OS — Hypervisor (KVM) that hosts the VM on the LC.

The Brocade device allows the interoperability between the SLX-OS CLI and the MMVM Linux shell if you are an SLX-OS user with admin permissions. As an admin user, you have the ability to perform the following tasks:

- Access the MMVM Linux shell from the SLX-OS CLI and run all commands supported by the Linux OS.
- Run the SLX-OS show and configuration commands directly from the MMVM Linux shell..
- Run Linux commands directly from the SLX-OS CLI.

   NOTE
   At the MMVM Linux shell, you have root level privileges.

Limitations

Consider these limitations when running commands from the MMVM Linux shell:

- Only the Bash shell is supported. The C shell and KornShell are not supported.
- You can install your favorite shell. However, shell activity logging is only supported for Bash.
- When you run Linux commands as part of the script or through a file, the device logs the script or file. The device does not log the commands.
- Pagination is not supported when you use the **cli_run** utility to execute SLX-OS CLI show commands from the shell.
- At the SLX-OS CLI, a window resizing issue occurs when you execute Linux commands such as **top**, **vim** using the **oscmd** command. We recommend that you execute these commands from the MMVM Linux shell.
- You can access the MMVM Linux shell from the SLX-OS CLI by using the **start-shell** command but it is only visible if you are logged in as an admin user.

NOTE
Inside the MMVM Linux shell, you have root level privileges and can run all supported Linux commands.

TPVM storage management

Consider the following storage management rules and restrictions when configuring TPVM.

Consider the number of disks that are manually or automatically created and their storage capacities:

- A 256 GB solid-state drive (SSD) is dedicated to the TPVM.
- A 32 GB boot disk (*vda*) that is automatically created when the TPVM is installed.
- You can manually create up to 3 virtual disks. The maximum total size of these disks cannot exceed 100 GB.

If the TPVM is uninstalled, the boot disk (*vda*) is deleted and all information on the boot disk is lost. However, the user created disks remain intact. You can use these disks to preserve data when you upgrade the TPVM or uninstall the TPVM.

There are some restrictions on removing disks.

- The last disk added must be removed first.
- You must remove disks in the reverse order of creation.
- You must unmount the disk before removing.

See the section Common TPVM commands on page 80.

TPVM console and management interface

Once TPVM is running, it can be accessed from a serial port and the management interface.

TPVM can be managed independently on both the MM1 and MM2. To access TPVM through a console port you press Ctrl-Y + 2.

The management IP address is automatically assigned on boot by DHCP.

- eh0 management interface uses DHCP mechanism
- eth1 insight interface uses Static IP Address assignment

You can also configure the TPVM Linux OS with your preferred method of IP address assignment. Manually assigning an IP address requires access to a console for the initial setup. A static IP address is configured by modifying the `/etc/network/interfaces` file, where you remove the DHCP and add static IP address information to a specific interface. See the section Assigning a static IP address on the TPVM Linux OS on page 90.

Kernel–based virtual machine

A kernel-based virtual machine (KVM) is a full virtualization structure for Linux on x86 base systems that turns the Linux kernel into a hypervisor.

KVM consists of a loadable kernel module, `kvm.ko`, that provides the core virtualization infrastructure and a processor specific module, either `kvm-intel.ko` or `kvm-amd.ko`.

By using KVM, you can run multiple VMs using Linux or Windows images. Each VM has private virtualized hardware: a network card, disk, graphics adapter, and so on.

KVM is open source software. The kernel component of KVM is included in Linux.

TPVM resource protection, isolation, and openness

Fully open non-proprietary platform

TPVM is based on Ubuntu 14.04 LTS 64 bit Linux 3.19. You can deploy container technologies (Dockers, Linux containers) in TPVM. We also bundle prequalified applications as part of TPVM.

See the section

CPU resource protection and isolation

Kernel-based Virtual machine (KVM) technology provides complete sandboxing of TPVM. TPVM does not affect the memory module VM (MMVM) operation or steal resources from the MMVM, nor do TPVM application or Linux kernel crashes affect the MMVM operation.

There are strict upper limits for CPU, memory, and disk space usage provided by TPVM. The resource limits are::

- Two CPU cores out of a total of 8 (4 physical cores × 2 hyperthreads per core = 8 cores).
- 4 GB of SDRAM (out of 16 GB total SDRAM).
- 32 GB of disk space (out of 256 GB total disk space).
- 10 GB insight interface port on each MM card is dedicated to TPVM.

TPVM troubleshooting commands

Use these Linux commands to check the status of TPVM and troubleshoot port mirroring on the TPVM.

TPVM status commands

To check the status of TPVM:

```
device# tpvmadm show
```

> **NOTE**
> The **tpvmadm** command is only available at the Linux device command prompt.

To check if the TPVM is installed on management module (MM)1:

```
device# tpvmadm show mm1
```

The **tpvmadm show disk all** command lists all the user created virtual disks.

```
device# tpvmadm show_disk all mm2
disk: vdb
Capacity: 40.00 GiB
Allocation: 136.01 MiB

disk: vdc
Capacity: 25.00 GiB
Allocation: 183.01 MiB

total:
Capacity: 100.00 GiB
Allocation: 65.00 GiB
Available: 35.00 GiB
```

> **NOTE**
> vda is the boot disk, created at installation.

> **NOTE**
> The full Linux **tpvmadm** command syntax canned be found in your Linux manual.

Display an Ethernet interface.

```
device# ifconfig eth1
   Bcast:127.2.255.255  Mask:255.255.0.0
        inet6 addr: feth1      Link encap:Ethernet  HWaddr 00:a0:c9:00:00:01
        inet addr:127.2.1.1 e80::2a0:c9ff:fe00:1/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:123252 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8682 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:174954039 (174.9 MB)  TX bytes:1794742 (1.7 MB)
```

> **NOTE**
> **ifconfig** is a command in the Linux OS (and others) for network interface configuration.

Display NIC parameters

```
device# ethtool eth1
Settings for eth1:
    Supported ports: [ FIBRE ]
    Supported link modes:   1000baseT/Full
                            10000baseT/Full
    Supported pause frame use: No
 Supports auto-negotiation: Yes
    Advertised link modes:  1000baseT/Full
                            10000baseT/Full
    Advertised pause frame use: Symmetric
    Advertised auto-negotiation: Yes
    Speed: 10000Mb/s
    Duplex: Full
    Port: Other
    PHYAD: 0
    Transceiver: external
    Auto-negotiation: on
    Supports Wake-on: umbg
    Wake-on: g
    Current message level: 0x00000007 (7)
                                  drv probe link
    Link detected: yes
```

> **NOTE**
> **ethtool** is a utility for Linux OS that queries or controls network driver and hardware settings.

```
device# ethtool -S eth1
NIC statistics:
    rx_packets: 153447
    tx_packets: 10035
    rx_bytes: 218685733
    tx_bytes: 1890188
    rx_errors: 0
    tx_errors: 0
    rx_dropped: 0
    tx_dropped: 0
    multicast: 22
    collisions: 0
rx_over_errors: 0
rx_crc_errors: 0
rx_frame_errors: 0
```

The -S flag specifies statistics (case sensitive).

Resource usage

Display all processes, their status, and resource usage.

```
device# ps aux | grep TPVM
```

> **NOTE**
> The aux parameter means a = show processes for all users, u = display the process's user/owner, x = show processes not attached to a terminal.

Display a list of the top users of your system's resources - CPU shares and memory.

```
device# top -p
```

Route commands

Display existing routes.

```
device# route
```

Display routes in numeric form.

```
device# route -n
```

Add a route.

```
device# route add -net Net-IP-Addr  netmask Net-IP-Addr-Mask gw Gw-IP
```

Add a default gateway.

```
device# route add default gw Gw-IP
```

> **NOTE**
> These are just ta few of the Linux commands that could help troubleshoot TPVM and port mirroring. Consult your Linux manual or online help for explanations of the commands and what flags are available.

TPVM upgrade

A TPVM upgrade is decoupled from that of the host device. It is treated as a server upgrade, where data generated in the previous version is preserved.

the following list three of the most likely use case scenarios:

1. Disruptive upgrade - TPVM kernel changes in new release, it is coupled with the device upgrade.

2. Non-disruptive upgrade - New applications are packaged in TPVM in new release, release and upgrade only packages.

3. Non-disruptive upgrade - Changes to current packages controlled by us (Brocade) in new release, release and upgrade only packages.

There are also security updates including scenarios that follow thee use-cases:

1. Security update to packages not controlled by Brocade - You can get direct updates from Ubuntu itself no need to go through Brocade.

2. Security update to packages controlled by Brocade - Scenario 3 above.

3. Security update to the kernel - Scenario 1 above.

Assigning a static IP address on the TPVM Linux OS

To use a static IP address, you add the static method for the interface in the file `/etc/network/interfaces`.

1. Get your current address, net mask, and broadcast.

    ```
    device# ifconfig
    ...
    eth0 Link encap:Ethernet HWaddr 00:0a:21:ff:45:2a
     inet addr:10.10.10.0 Bcast:10.10.10.255 Mask:255.255.255.0
    ...
    ```

    The above example displays the first Ethernet interface identified as eth0.

2. Get your gateway and network

    ```
    device# route -n
    Kernel IP routing table
     Destination  Gateway       Genmask        Flags  Metric  Ref  Use  Iface
     0.0.0.0      10.10.10.2    0.0.0.0        UG     100     0    0    eth0
     172.16.77.0  0.0.0.0       255.255.255.0  U      0       0    0    eth0
    ```

    Use flags u and g for the route gateway. The other IP address is the network IP.

3. Open the interfaces file.

    ```
    device# sudo nano /etc/network/interfaces
    ```

    Nano is the GNU version of the Pico text editor, use the editor of your choice.

4. Find the DHCP settings in the `interfaces` file, they will appear as text similar to the following:

    ```
    ...
    auto eth0
    iface eth0 inet dhcp
    ...
    ```

5. Replace the text seen in step 4 with the following.

    ```
    ...
    auto eth0
    iface eth0 inet static
    address 10.0.0.100
    netmask 255.255.255.0
    gateway 10.0.0.0
    ...
    ```

    The above example configures the first Ethernet interface identified as eth0.

6. Save the file and exit to the command prompt.

7. Make sure your name server IP address is your gateway IP address.

    ```
    device# sudo nano /etc/resolv.conf
    ```

8. Restart the networking components.

    ```
    device# sudo /etc/init.d/networking restart
    ```

9. If you want this as a permanent change, remove the DHCP client so it can no longer assign dynamic IP addresses.

    ```
    device# sudo apt-get remove dhcp-client
    ```

10. Verify connectivity.

    ```
    device# ping www.brocade.com
    ```

11. Manually enable the interface.

```
device# sudo ifup eth0
```

Insight interface overview

The insight interface is the port(s) that connects the line card (LC) packet processors and the MM BP switches.

The insight interface provides data plane access for the TPVM running on both of the MM1 and MM2 CPUs.

Insight interface supported features

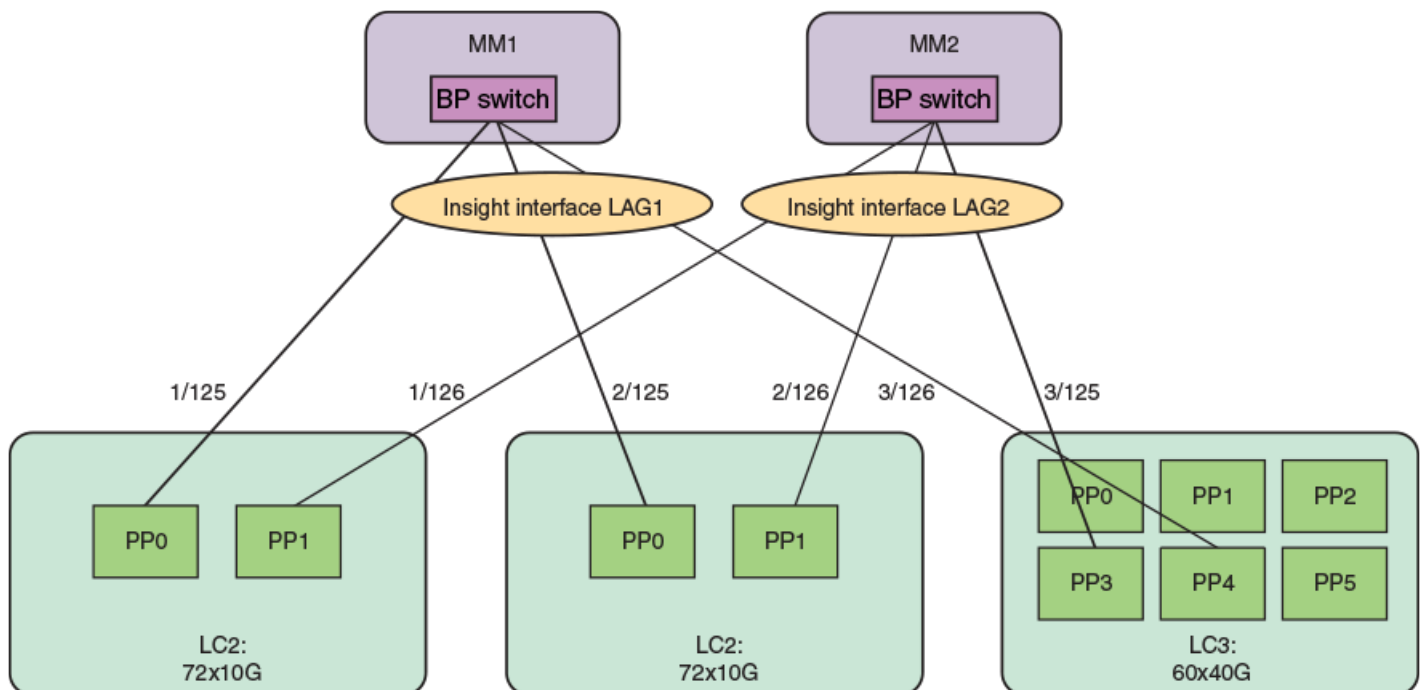Insight interface port-channel supports the following third-party features and hardware.

Insight interface supports the standard features seen in other front end interfaces including:

- Port and ACL-based mirroring destination
- QoS and rate shaping

Insight interface internal connectivity

LAG based modeling handles the failover of a linear program (LP) module.

FIGURE 5 High level view of insight interface ports and port-channels

Insight interface data path

The insight interface port is provided as a pre-existing port-channel interface to which all the insight interfaces on the LC are automatically added during system boot.

The port-channel 1 interface and the port-channel 2 interface are created as default LAGs in the system for MM1 and MM2 respectively. These are visible to you, configured with default settings, and are static LAGs only. All other options on the LAG are disabled. Insight interface ports and port-channels work independently with each providing up to 20 GB bandwidth for applications running on MM1 and MM2.

Insight interface port-channel creation, addition, or deletion is similar to the standard port-channel creation, addition, or deletion except it is programmatically invoked during system initialization and the insight interfaces on the LC are automatically and statically associated with he insight interface port-channel on the MM.

For example, when the LC in slot 1 comes online, the insight interface LAG port on the LC `interface ethernet 1/125` is statically associated with the insight interface LAG `interface port-channel 1` on MM1. Similarly `interface ethernet 1/126` is statically associated with the insight interface LAG `interface port-channel 2` on MM2.

Insight interface physical ports

There are two insight interface ports per line card (LC).

One of the insight interface ports is connected to a TPVM on Management Module (MM) 1 and the other is hard-wired to the TPVM port on MM2.

TABLE 6 Insight interface ports mapping:

LC type	LC port	LC tower	Destination
36x100 GB	1/125	3	MM1
	1/126	4	MM2
72x10 GB	1/125	0	MM1
	1/126	1	MM2

Consider the following when configuring insight interface:

- Link detection and reporting are the same as in regular front ports.
- The regular front ports are in an administratively down state by default. However, the insight interface ports are in an administratively up state.
- Forward error correction (FEC) between the media-independent interface (MII) and physical medium attachment (PMA) layers is enabled.
- 10 GB KR auto negotiation is enabled on BP switches and line card packet processor ports for insight interface ports.

Configure inbound ACL-based mirroring to the insight interface

A Layer 2 or Layer 3 extended ACL permit filter must be configured to mirror the incoming matching traffic to a given port. You can also configure different mirror ports for different filters in the same ACL.

Enabling ACL-based port mirroring

Follow these high level steps to enable ACL-based port mirroring.

1. Create an ACL.
   - Traffic can only be selected using a permit clause.
   - The ACL can be bound to a physical pot or a LAG.
   - The physical port or LAG interface should be configured as a switchport.
   - Configure the **mirror** keyword in an ACL filter to enable inbound ACL mirroring. This directs selected traffic to the mirrored port.
2. Associate the ACL mirror source and destination port. The mirror source port should be physical and the mirror destination port is either a physical port or a LAG port.
3. Bind the ACL to an interface.
4. Save the configuration.

Configuring inbound ACL-based mirroring to the insight interface

Follow these steps to configure inbound ACL-based mirroring.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create an extended Layer 2 ACL.

   ```
   device(config)# mac access-list extended macl
   ```

3. Configure the Layer 2 ACL for mirroring.

   ```
   device(conf-macl-ext)# seq 10 permit host 0010.9400.0010 host 0010.9400.0014 vlan 20 count mirror
   device(conf-macl-ext)# seq 20 permit host 0010.9400.0011 host 0010.9400.0015 vlan 20 count mirror
   device(conf-macl-ext)# seq 30 permit host 0010.9400.0012 host 609c.9f01.58cb vlan 21 count mirror
   device(conf-macl-ext)# seq 40 permit host 0010.9400.0013 host 609c.9f01.58cb vlan 22 count mirror
   device(conf-macl-ext)# seq 50 permit any any count mirror
   ```

4. Return to global configuration mode.

   ```
   device(conf-macl-ext)# exit
   ```

5. Create an extended IPv4 ACL.

   ```
   device(config)# ip access-list extended ipv4acl
   ```

6. Configure the IPv4 ACL for mirroring.

   ```
   device(conf-ipv4acl-ext)# seq 10 permit ip host 11.12.13.14 any count mirror
   ```

7. Return to global configuration mode.

   ```
   device(conf-ipv4acl-ext)# exit
   ```

8. Associate the ACL destination mirror port.

   ```
   device(config)# acl-mirror source ethernet 4/1 destination port-channel 1
   ```

9. Enter configuration mode for the source mirror port.

```
device(config)# interface ethernet 1/4
```

10. Bind the Layer 3 IP ACL to the source mirror port.

    a) Bind the Layer 2 ACL to the source mirror port.

    ```
    device(conf-if-eth-4/1)# mac access-group mac1 in
    ```

    b) Bind the IPv4 ACL to the source mirror port.

    ```
    device(conf-if-eth-4/1)# ip access-group ipv4acl in
    ```

11. Return to privileged exec mode.

```
device(conf-if-eth-4/1)# end
```

12. Verify the configuration.

```
device# show statistics access-list interface ethernet 4/1 in
mac access-list macl on Ethernet 4/1 at Ingress (From User)
    seq 10 permit host 0010.9400.0010 host 0010.9400.0014 vlan 20 count mirror (105555094236 frames)
    seq 20 permit host 0010.9400.0011 host 0010.9400.0015 vlan 20 count mirror (105555103123 frames)
    seq 30 permit host 0010.9400.0012 host 609c.9f01.58cb vlan 21 count mirror (105555072247 frames)
    seq 40 permit host 0010.9400.0013 host 609c.9f01.58cb vlan 22 count mirror (105555083432 frames)
    seq 50 permit any any count mirror (0 frames)
```

13. Save the configuration.

```
device# copy running-config startup-config
```

Inbound ACL-based mirroring to the insight interface configuration example (Layer 2)

```
device# configure terminal
device(config)# mac access-list extended macl
device(conf-macl-ext)# seq 10 permit host 0010.9400.0010 host 0010.9400.0014 vlan 20 count mirror
device(conf-macl-ext)# seq 20 permit host 0010.9400.0011 host 0010.9400.0015 vlan 20 count mirror
device(conf-macl-ext)# seq 30 permit host 0010.9400.0012 host 609c.9f01.58cb vlan 21 count mirror
device(conf-macl-ext)# seq 40 permit host 0010.9400.0013 host 609c.9f01.58cb vlan 22 count mirror
device(conf-macl-ext)# seq 50 permit any any count mirror
device(conf-macl-ext)# exit
device(config)# acl-mirror source ethernet 4/1 destination port-channel 1
device(config)# interface ethernet 1/4
device(conf-if-eth-4/1)# mac access-group mac1 in
device(conf-if-eth-4/1)# end
device# show statistics access-list interface ethernet 4/1 in
device# copy running-config startup-config
```

> **NOTE**
> Only the Layer 2 ACL creation is shown in this example.

Configure port-based mirroring to the insight interface

Port mirroring is used to analyze, debug, or diagnose errors on a network.

If port mirroring is enabled, the device sends a copy of network packets from one switchport to a monitoring connection on another switchport. This is commonly used for network appliances that require traffic monitoring.

Configuring port-based mirroring to the insight interface

Follow these steps to enable a port-mirroring session for monitoring traffic.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter monitor session configuration mode to enable a port-mirroring session for monitoring traffic..

   ```
   device(config)# monitor session 1
   ```

   Monitoring session IDs can range from 1 to 512.

3. Configure the monitoring session for ingress traffic.

   ```
   device(config-session-1)# source ethernet 4/1 destination port-channel 1 direction rx
   ```

   In this example the ingress traffic is mirrored from source port 1/1 to port-channel 1.

4. Got to monitoring session 2.

   ```
   device(config-session-1)# monitor session 2
   ```

5. Configure the monitoring session for egress traffic.

   ```
   device(config-session-2)# source ethernet 4/2 destination port-channel 1 direction tx
   ```

6. Got to monitoring session 3.

   ```
   device(config-session-2)# monitor session 3
   ```

7. Configure the monitoring session for both ingress and egress traffic.

   ```
   device(config-session-3)# source ethernet 4/3 destination port-channel 1 direction both
   ```

8.

9. Return to privileged exec mode.

   ```
   device(config-session-1)# end
   ```

10. Verify the configuration

    ```
    device# show monitor session 1
    Session                : 1
    Description            : [None]
    State                  : Enabled
    Source Interface       : Eth 4/1 (Up)
    Destination Interface  : Po 1 (Up)
    Direction              : Rx
    ```

    The Enabled State confirms that port mirroring on the line card packet processor is successfully configured. Execute this **show** command for each monitoring session.

11. Save the configuration.

    ```
    device# copy running-config startup-config
    ```

Port-based mirroring to the insight interface configuration example

```
device# configure terminal
device(config)# monitor session 1
device(config-session-1)# source ethernet 4/1 destination port-channel 1 direction rx
device(config-session-1)# monitor session 2
device(config-session-2)# source ethernet 4/2 destination port-channel 1 direction tx
device(config-session-2)# monitor session 3
device(config-session-3)# source ethernet 4/3 destination port-channel 1 direction both
device(config-session-1)# end
device# show monitor session 1
device# copy running-config startup-config
```

Insight interface port-channel

The insight interface port is provided as a pre-existing port-channel interface.

Insight interfaces are automatically created on each LC to be part of the default LAG. Insight interfaces are part of the shipped configuration; you cannot change the configuration.

The insight interface appears as such in the running configuration:

```
device# show running-config interface ethernet 1/125
interface Ethernet 1/125
 description Insight port to MM1
 channel-group 1 mode on type standard
 no shutdown
 !
...
device# show running-config interface ethernet 1/126
interface Ethernet 1/126
 channel-group 2 mode on type standard
 description Insight port to MM2
 no shutdown
!
…
```

> **NOTE**
> The Ethernet port 1/125 is the line card packet processor port associated with MM1.
> The Ethernet port 1/126 is the line card packet processor port associated with MM2.

Only one of the LC insight interfaces associated with port-channel is enabled. The insight interface port management component processes the LC UP or DOWN events, the management component also maintains list of member ports for the UP line card. Whenever a given selected LC goes down, the next insight interface LC port is enabled without bringing down the LAG interface.

All LCs must be down for the LAG to be disabled. When the first LC comes up, the LAG is enabled and that LC port is used as the active port in the LAG.

Insight interface port-channel limitations and restrictions

When configuring the insight interface port-channel consider the following limitations and restrictions:

- Configuring the insight interface port-channel must be done under the interface port-channel mode.
- Only two insight interface port-channel interfaces are automatically created on boot, one for each MM. These are port-channel 1 and port-channel 2.
- The LC connects slot/125 and slot/126 to MM1 and MM2 respectively.
- Physical port association to a port-channel is not configurable. It is automatically done on boot.
- Only one physical port within the port-channel is enabled, per Insight Interface, at a given time.

- Configurations under **interface ethernet** *slot/125* or *slot/126* are not supported in this release.
- Insight interface port-channels cannot be deleted.
- The maximum bandwidth is 10 Gbps.

Insight interface LC recovery

During link aggregation group (LAG) member failover, when the primary LC is powered down:

- There is a disassociation of the primary insight interface from the insight interface port-channel.
- There is an automatic selection of another insight interface LC as the primary insight interface port-channel.
- The port-channel stays up during the LC transition.

Insight interface traffic management and QoS

From a traffic management perspective, QoS for insight interface is similar to QoS for a regular port. The differences is the QoS configuration is applied to an insight interface LAG (port-channel) instead of an interface.

Consider the following when you configure this feature:

- The TM supports egress scheduling, rate shaping, WRED, and ingress buffer management for insight interface.
- TM egress scheduling and shaping for the insight interface must be configured under a port-channel interface.
- The QoS configuration is automatically applied to all ports in the port-channel.
- Follow the same configuration procedures for ingress buffer management and WRED as you would with a standard port.

QoS egress scheduling

QoS egress scheduling works under the following rules, restrictions, and limitations:

- You must use a credit request/grant mechanism to perform egress scheduling QoS.
- The maximum credit size is1024 Bytes.
- For each egress port there are 8 Virtual Output Queues (VOQs) allocated on each ingress transmit module (TM) core to support 8 priorities.
- Egress scheduling supports strict priority (SP), weighted fair queue (WFQ), and mixed mode scheduling.
- You can specify weighted for each VOQ only in WFQ mode.
- Fair queue (FQ) scheduling between VOQs from different TMs and with the same priority is permitted.

The figure below illustrates a QoS egress scheduling scheme.

FIGURE 6 QoS egress scheduling scheme
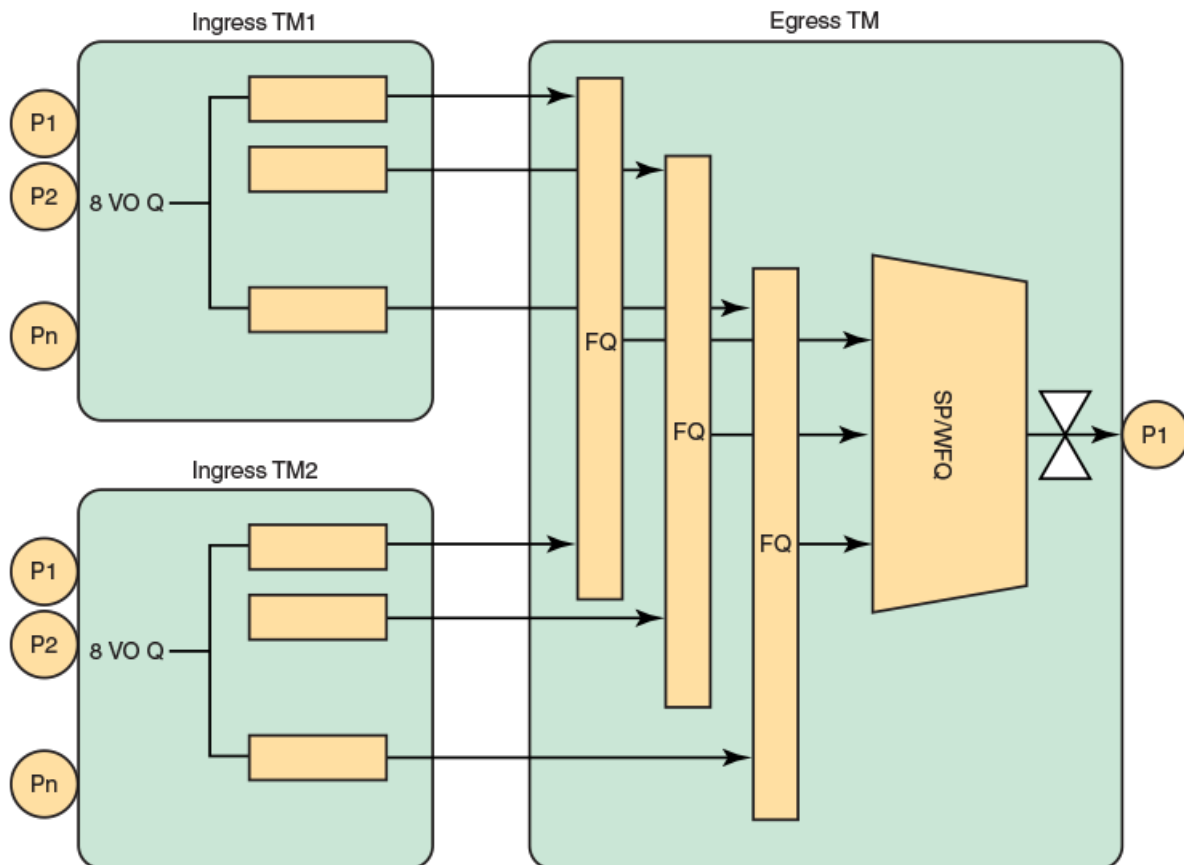
In the figure above P1 is the insight interface.

Configuring QoS egress scheduling

Follow the below steps to configure QoS egress scheduling.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode for port-channel 1.

```
device(config)# interface port-channel 1
```

3. Specify the option for strict priority mode to determine strict priority queues.

```
device(port-channel-1)# qos queue scheduler strict-priority 4
```

   There are seven traffic classes. Specify the weight for the priority If the priority is in WFQ mode.

4. Return to privileged exec mode.

```
device(port-channel-1)# end
```

5. Verify the configuration.

    ```
    device# show qos interface port-channel 1
    ```

6. View the VOQ statistics.

```
device# show tm voq-stat ingress-device ethernet 1/15 egress-port ethernet 1/125

VOQ-Counters:
=================================================

Priority 0
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 1
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 2
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 3
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 4
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 5
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 6
-----------------------------------------------
EnQue Pkt Count                            0
EnQue Bytes Count                          0
Total Discard Pkt Count                    0
Total Discard Bytes Count                  0
Current Queue Depth                        0
Maximum Queue Depth since Last read        0

Priority 7
```

```
 --------------------------------------------
 EnQue Pkt Count                           0
 EnQue Bytes Count                         0
 Total Discard Pkt Count                   0
 Total Discard Bytes Count                 0
 Current Queue Depth                       0
 Maximum Queue Depth since Last read       0
```

7. Save the configuration.

```
device# copu running-config startup-config
```

QoS egress scheduling configuration example

```
device# configure terminal
device(config)# interface port-channel 1
device(port-channel-1)# qos queue scheduler strict-priority 4
device(port-channel-1)# end
device# show qos interface port-channel 1
device# show tm voq-stat ingress-device ethernet 1/15 egress-port ethernet 1/125
device# copy running-config startup-config
```

QoS rate shaping

QoS rate shaping allows you to limit egress traffic to a specified rate.

Rate shaping works under the following rules and limitations:

*   If there is a higher data rate than the configured shaping rate, traffic is kept at the ingress VOQ.

*   The ingress TM tail drops packets if the queue is full.

*   Accuracy is +- 3%.

*   Configuration granularity is 1KB.

Troubleshooting port-mirroring

Follow these high level steps to troubleshoot port-mirroring.

1. MAC counters on source and destination interfaces can be verified by running the command : **show interface ethernet** *slot/port*.

    > NOTE
    > This command is specific to the LC, it can be executed only on an LC console and not on a MM.

2. To see if there are any packet drops, check the PP counters by running the command: **show ppc statistics device** *slot'port*.

    > NOTE
    > This command is specific to the LC, it can be executed only on an LC console and not on a MM.

3. To see if packets are sent to a destination queue, VOQ counters can be verified by running the command: **show tm voq-stat destination** *slot'port*.

4. MMVM commands include:

    a. **show interface port-channel [1 | 2]** (MM1) or 2 (MM2)

    b. **show port-channel**

    c. **show interface stats [brief | detail]**

    d. **show interface ethernet** *slot/125* (MM1)

    e. **show interface ethernet** *slot/126* (MM2)

Troubleshooting port-mirroring

Use these example in debugging port mirroring.

Configuring port mirroring from interface 1/2 to 1/1.

```
evice(config)# monitor session 1
device(config-session-1)# source ethernet 1/2 destination ethernet 1/1 direction rx
```

Display the MAC counters on the ingress interface:

```
device# show interface ethernet 1/2
Ethernet 1/2 is up, line protocol is up (connected)
Receive Statistics:
    1000 packets, 128000 bytes
    Unicasts: 1000, Multicasts: 0, Broadcasts: 0
...
(Output is truncated)
```

Display the MAC counters on the mirrored interface:

```
device# show interface ethernet 1/1 (Output is trimmed for brevity)
Ethernet 1/1 is up, line protocol is up (connected)
Transmit Statistics:
    1000 packets, 128000 bytes
    Unicasts: 1000, Multicasts: 0, Broadcasts: 0
    Underruns: 0
...
(Output is truncated)
```

Display the destination virtual output queuing (VOQ) counters:

```
device# show tm voq-stat destination 1/1
VOQ-Counters:
Priority 1:
EnQue Pkt Count                 1000
EnQue Bytes Count               148000
Total Discard Pkt Count               0
Total Discard Bytes Count             0
```

> **NOTE**
> This command can be used to check if the packets are received by the destination VOQ.

SLX-OS VM commands

Use these commands on the Brocade device to help troubleshoot the VM.

Display general interface information

```
device# show interface stats brief

              Packets       Error      Discards      CRC
```

```
Interface     rx           tx   rx           tx  rx          tx   rx
==========  ============ ============ ============ ============  ====
Po  1          16           2   0            0   0           0    0
Po  2           0           0   0            0   0           0    0
Eth  1/1        0           0   0            0   0           0    0
Eth  1/125      8           3   0            0   0           0    0
Eth  1/126      0           0   0            0   0           0    0
```

Display port-channel information

```
device# show interface port-channel 1
Port-channel 1 is up, line protocol is up
Hardware is AGGREGATE, address is 748e.f88f.5ffd
    Current address is 748e.f88f.5ffd
Description: Insight port-channel on MM1
Interface index (ifindex) is 671088641
Minimum number of links to bring Port-channel up is 1
MTU 2500 bytes
LineSpeed Actual     : 10000 Mbit
Allowed Member Speed : 10000 Mbit
Priority Tag disable
IPv6 RA Guard disable
Last clearing of show interface counters: 03:43:52
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 03:43:52

device# show interface port-channel 2
Port-channel 2 is up, line protocol is down (link protocol down)
Hardware is AGGREGATE, address is 0027.f817.1438
    Current address is 0027.f817.1438
Description: Insight port-channel on MM2
Interface index (ifindex) is 671088642
Minimum number of links to bring Port-channel up is 1
MTU 2500 bytes
LineSpeed Actual     : Nil
Allowed Member Speed : 10000 Mbit
Priority Tag disable
IPv6 RA Guard disable
Last clearing of show interface counters: 03:45:17
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
```

```
     Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
     Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 03:45:18

device# show running-config interface port-channel 1
interface Port-channel 1
 no vlag ignore-split
 description Insight port on MM1
 no shutdown

device# show running-config interface port-channel 2
interface Port-channel 2
description Insight port on MM1
 shutdown

device# show port-channel
 Static Aggregator: Po 1
 Aggregator type: Standard
    Eth 1/125
    Eth 4/125
 Static Aggregator: Po 2
 Aggregator type: Standard
    Eth 1/126
    Eth 4/126
```

Optional keywords are **summary**, **detail**, and **load-balance**.

Display Ethernet interface information

```
device# show interface ethernet 1/125
Ethernet 1/125 is up, line protocol is down (link protocol down)
Hardware is Ethernet, address is 0027.f817.12fe
    Current address is 0027.f817.12fe
Pluggable media not present
Interface index (ifindex) is 4704206921
MTU 2500 bytes
LineSpeed Actual      : Nil
LineSpeed Configured : Auto, Duplex: Full
Priority Tag disable
IPv6 RA Guard disable
Last clearing of show interface counters: 03:46:22
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 03:46:22

device# show interface ethernet 1/126
Ethernet 1/126 is up, line protocol is down (link protocol down)
Hardware is Ethernet, address is 0027.f817.12ff
    Current address is 0027.f817.12ff
Pluggable media not present
Interface index (ifindex) is 4704239690
MTU 2500 bytes
LineSpeed Actual      : Nil
LineSpeed Configured : Auto, Duplex: Full
Priority Tag disable
```

```
IPv6 RA Guard disable
Last clearing of show interface counters: 03:47:28
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 03:47:27
```

Password Recovery

Recovering the admin password from the root account

If you lose access to the SLX-OS admin account but you have access to the root account for the device, you can recover the password.

Perform the following steps to reset the admin password from the root account.

1. Open a session to access the device.

2. Log in as root.

3. Start the SLX-OS CLI.

   ```
   [root@device]# slxcli
   device#
   ```

4. Access global configuration mode.

   ```
   device# configure terminal
   ```

5. Reset the admin password.

   ```
   device(config)# username admin password password
   ```

   In this example, the admin password is reset to the default value of password.

   You can now use the admin account to manage the admin and user passwords by using normal password-management procedures.

VM root password recovery for Brocade SLX 9850

By default, the root account on the virtual machine (VM) for Brocade SLX 9850 is disabled. To log into root, you can log into the SLX-OS CLI and enable the root account from global configuration mode by using **root enable** command. In rare cases, SLX-OS CLI may not be available to enable the root account.

The ability to enable the root account and recover the root credentials (password) depends on the uboot environment variable. When the variable is set, it executes the root recovery logic based on the parameter set. The variable is not preserved across reboot. Every time a reboot occurs, the root account is disabled by default and this variable has to be set again to enable it unless the root account was not enabled from global configuration mode.

The root account access availability determines the method for password recovery:

- When the root account is disabled and the SLX-OS CLI is not available, you must recover the root login account. The password is also recovered.

- When the root account is enabled but you forget the password, you must recover the VM root password.

NOTE
The default password for the root account on the VM is
fibranne.

Recovering the VM root login account

When you need to recover the VM root password, and the root account is disabled and SLX-OS CLI is not available, you must recover
the VM root login account.

- To perform the recovery process on the active or standby management module (MM), you must have access to the console
  prompt.
- To perform the process on line cards (LCs), you can either use the direct console access if present or use remote console from
  the MM to access the LC console.

For VM root login recovery, perform the following steps.

1. Reboot the active MM, standby MM, or LC for which you are trying to recover the root credentials.

   ```
   # reboot
   Press Esc during reboot.
   Hit ESC to stop autoboot:  0
   FPGA f6000720 -> 0x12

   1) Start system.
   2) Recover password.
   3) Enter command shell.

   Option?
   ```

2. Choose option 3 to access the uboot prompt.

   ```
   Option? 3
   =>
   ```

   The uboot prompt appears.

3. Define the root login value for the root recover environment variable.

   ```
   => setenv VM_Root_Recover RootLogin
   =>
   ```

   This step sets the VM_Root_Recover variable with the RootLogin value.

4. Save the variable to flash memory.

   ```
   => saveenv
   Saving Environment to SPI Flash...
   SF: Detected W25Q128BV @ 0:0 with page size 256 Bytes, erase size 64 KiB, 32 KiB, 4 KiB, total 16 MiB
   Erasing SPI flash...Writing to SPI flash...
   Erasing SPI flash...Writing to SPI flash...done
   =>
   ```

5. Boot the MM or LC.

   ```
   => boot
   6912784 bytes read in 152 ms (43.4 MiB/s)
   Valid Boot Flag
   Setup Size = 0x00004400
   Magic signature found
   Using boot protocol version 2.0c
   Linux kernel version 3.14.17 (raop@hq1-ub-ecbld-373) #1 SMP Thu Jul 7 19:43:15 UTC 2016
   ```

   The root account is now enabled. You can login with the default password.

If the SLX-OS CLI is available, you can recover the the active MM root account by using the SLX-OS CLI **root enable** command.

Recovering the VM root password

If you forgot the password for the VM root account, you can recover the default password.

- To perform the recovery process on the active or standby management module (MM), you must have access to the console prompt.
- To perform the process on line cards (LCs), you can either use the direct console access if present or use remote console from the MM to access the LC console.

For VM root password recovery, perform the following steps.

1. Reboot the active MM, standby MM, or LC for which you are trying to recover the root credentials.

   ```
   # reboot
   Press Esc during reboot.
   Hit ESC to stop autoboot:  0
   FPGA f6000720 -> 0x12

   1) Start system.
   2) Recover password.
   3) Enter command shell.

   Option?
   ```

2. Choose option 3 to access the uboot prompt.

   ```
   Option? 3
   =>
   ```

3. Define the root password value for the root recovery environment variable.

   ```
   => setenv VM_Root_Recover RootPasswd
   =>
   ```

   This step sets the VM_Root_Recover variable with the RootPasswd value.

4. Save the variable to flash memory.

   ```
   => saveenv
   Saving Environment to SPI Flash...
   SF: Detected W25Q128BV @ 0:0 with page size 256 Bytes, erase size 64 KiB, 32 KiB, 4 KiB, total 16 MiB
   Erasing SPI flash...Writing to SPI flash...
   Erasing SPI flash...Writing to SPI flash...done
   =>
   ```

5. Boot the MM or LC.

   ```
   => boot
   6912784 bytes read in 152 ms (43.4 MiB/s)
   Valid Boot Flag
   Setup Size = 0x00004400
   Magic signature found
   Using boot protocol version 2.0c
   Linux kernel version 3.14.17 (raop@hq1-ub-ecbld-373) #1 SMP Thu Jul 7 19:43:15 UTC 2016
   ```

   You can now login as root with the default password of fibranne.

Brocade SLX-OS Management Configuration Guide, 16r.1.01
53-1004819-02   109

SLX-OS and Linux Shell Interoperability

Overview

The Brocade device supports interoperability between the SLX-OS CLI and the management module VM (MMVM) Linux shell.

As an SLX-OS user with admin permissions, you can perform the following tasks:

* Access the MMVM Linux shell from the SLX-OS CLI and and run all commands supported by the Linux OS.

* Run the SLX-OS **show** and configuration commands directly from the MMVM Linux shell.

* Run Linux commands and user-defined scripts directly from the SLX-OS CLI.

  NOTE
  At the MMVM Linux shell, you have root level privilege.

Limitations

* Only the Bash shell is supported. The C shell and KornShell are not supported.

* You can install your favorite shell, However, shell activity logging is not supported for other shells.

* When you run Linux commands as part of the script or through a file, the device logs the script or file execution. It does not log the commands.

* When you use the cli_run utility to execute SLX-OS CLI **show** commands from the shell, pagination is not supported.

* At the SLX-OS CLI, a window resizing issue occurs when you execute Linux commands such as **top** using the **oscmd** command. Brocade recommends that you execute these commands from the MMVM Linux shell.

* If you open multiple Bash sessions, the Linux shell timeout is applicable only on the current Bash session.

* When you use the **oscmd** command, the execution of the following Linux commands are not supported:

  - **bash**
  - **chgrp**
  - **chmod**
  - **chown**
  - **cp**
  - **kill**
  - **killall**
  - **mv**
  - **rm**
  - **rmdir**
  - **sh**

- **script**
- **vi**
- **vim**

Brocade recommends that you execute these Linux commands from the MMVM Linux shell.

- At the MMVM Linux shell, do not use the **passwd** command to change a non-root user password. Brocade recommends that you can use the **username** command from the SLX-OS CLI for the modification of user accounts. For information on modifying a user account, see the *Brocade SLX-OS Security Configuration Guide*.

Accessing the Linux shell from the SLX-OS CLI

With the admin role, you can access the MMVM Linux shell from the SLX-OS CLI by using the **start-shell** command.

> **NOTE**
> The **start-shell** command is only visible in the CLI when you are configured as a user with the admin role.
> Inside the MMVM Linux shell, you will have root level privilege and can run all supported Linux commands.

Perform the following steps to access the shell.

1. From privileged EXEC mode, access the MMVM Linux shell.

```
device# start-shell
Entering Linux shell for the user: admUser

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[admUser@SLX]#
```

You can also run this command from global configuration mode by entering the **do start-shell** command.

> **NOTE**
> The idle timeout of the MMVM Linux shell session is five minutes. If the MMVM Linux shell is idle for more than five minutes, you are automatically logged out of Linux shell and returned to SLX-OS CLI.

2. Exit the shell and return to the SLX-OS CLI.

```
[admUser@SLX]# exit
```

Upon exiting, the following message appears and you return to the SLX-OS CLI prompt.

```
exit

Exited from Linux shell

device#
```

Executing SLX-OS commands from the Linux shell

From the MMVM Linux shell, you can directly execute a single SLX-OS CLI privileged EXEC command or multiple commands in a file.

When the SLX-OS CLI commands are in a file, separate each command with a new line.

Perform the following steps to execute SLX-OS commands from the shell.

1. Execute a single SLX-OS privileged EXEC command.

   ```
   [admUser@SLX]# cli_run -c "show ip interface brief" | grep Port-channel > /tmp/interface
   ```

   The **cli_run -c** utility executes a single SLX-OS CLI command. In this example, the SLX-OS CLI command output is redirected to the /tmp/interface file using standard Linux filters.

2. Display the contents of the file to verify the redirection.

   ```
   [admUser@SLX]# cat /tmp/interface
   Port-channel 1          unassigned                              administratively down    down
   Port-channel 2          unassigned                              administratively down    down
   ```

3. Execute multiple SLX-OS CLI commands from a file.

   ```
   [admUser@SLX]# cli_run -f slxcli_cmd_file
   ```

   The **cli_run -f** utility executes multiple SLX-OS CLI commands in file. In this example, the slxcli_cmd_file contains the following commands that are executed.

   ```
   [admUser@SLX]# cat slxcli_cmd_file
      show ssh server status
      conf t
      router mpls
      lsp test1
      from 10.2.3.4
   ```

4. Execute multiple SLX-OS CLI commands in a file and redirect the output of the executed commands to another file.

   ```
   [admUser@SLX]# cli_run -f slxcli_cmd_file > newfile
   ```

   In this example, the contents of the slxcli_cmd_file is executed and the newfile file contains the redirected output.

5. Display the contents of the file to verify that it contains the redirected output.

   ```
   [admUser@SLX]# cat newfile
      Welcome to the Brocade SLX-OS Software
      admUser connected from 127.0.0.1 using console on SLX
      device# show ssh server status | nomore
      VRF-Name: mgmt-vrf      Status: Enabled
      VRF-Name: default-vrf   Status: Enabled
      device# conf t
      Entering configuration mode terminal
      device(config)# router mpls
      device(config-router-mpls)# lsp test1
      device(config-router-mpls-lsp-test1)# from 10.2.3.4
   ```

Executing Linux shell commands from SLX-OS

With the admin role, you can execute Linux commands directly from SLX-OS CLI.

To execute a Linux command, enter the **oscmd** command followed by the Linux command.

> **NOTE**
> The **oscmd** command is only visible in the CLI when a user has been configured with the admin role.

```
device#  oscmd ps -ef
UID         PID  PPID  C STIME TTY          TIME CMD
root          1     0  0 Jul24 ?        00:00:04 /sbin/init
root          2     0  0 Jul24 ?        00:00:00 [kthreadd]
```

```
root          3       2   0 Jul24 ?        00:00:00 [migration/0]
root          4       2   0 Jul24 ?        00:00:03 [ksoftirqd/0]
root          5       2   0 Jul24 ?        00:00:00 [migration/1]
root          6       2   0 Jul24 ?        00:00:03 [ksoftirqd/1]
root          7       2   0 Jul24 ?        00:00:00 [migration/2]
root          8       2   0 Jul24 ?        00:00:02 [ksoftirqd/2]
root          9       2   0 Jul24 ?        00:00:00 [migration/3]
root         10       2   0 Jul24 ?        00:00:02 [ksoftirqd/3]
root         11       2   0 Jul24 ?        00:00:00 [migration/4]
root         12       2   0 Jul24 ?        00:00:02 [ksoftirqd/4]
root         13       2   0 Jul24 ?        00:00:00 [migration/5]
root         14       2   0 Jul24 ?        00:00:03 [ksoftirqd/5]
root         27       2   0 Jul24 ?        00:00:00 [cpuset]
root         28       2   0 Jul24 ?        00:00:01 [khelper]
root         31       2   0 Jul24 ?        00:00:00 [netns]
root         34       2   0 Jul24 ?        00:00:00 [async/mgr]
root        270       2   0 Jul24 ?        00:00:00 [sync_supers]
root        272       2   0 Jul24 ?        00:00:00 [bdi-default]

...

root     8kblockd/6]182    1   0 Jul24 ?        00:00:00 /usr/sbin/inetd
root     8237    1   0 Jul24 ?        00:00:00 /usr/sbin/sshd
admin    27536 27535   0 04:19 pts/4    00:00:00 ps -ef
```

In this example, the Linux **ps -ef** command lists the process status.

Executing user-defined scripts from the SLX-OS CLI

The SLX-OS CLI allows you to execute user-defined scripts that you have either copied from a network location to the flash memory of the SLX-OS device, or created in the MMVM Linux shell of the SLX-OS device.

Downloading a script to the SLX-OS device

After writing and testing a user-defined script file, you can copy it from an accessible network location to the flash memory of the SLX-OS device, similar to the following example.

```
device# copy ftp://MyUserID:MyPassword@10.10.10.10//<copy_script.sh> flash://copy_script.sh
```

> **NOTE**
> To copy a script from a USB device, refer to "Copying a file from a USB
> device."

After copying the script to the device, verify that the script file is displayed with the list of files in the flash memory of the device.

```
device# dir
total 24
drwxr-xr-x  2 root     sys       4096 Oct 26 15:22 .
drwxr-xr-x  3 root     root      4096 Oct  1  1970 ..
-rw-r--r--  1 root     root      1051 Oct 24 16:09 copy_script.sh
-rw-r--r--  1 root     root       207 Oct 24 16:09 create_vlans.py
-rw-r--r--  1 root     sys        557 Oct 26 10:37 defaultconfig.novcs
-rw-r--r--  1 root     sys        778 Oct 26 10:37 defaultconfig.vcs

1922789376 bytes total (828317696  bytes free)
```

If the copied script does not have executable permission, you need to assign executable permission from the Linux shell.

NOTE

To access the MMVM Linux shell, refer to Accessing the Linux shell from the SLX-OS CLI on page 66.

```
[admUser@SLX]#cd /var/config/vcs/scripts/

[admUser@SLX]#chmod 755 copy_script.sh

[admUser@SLX]#ls -lart copy_script.sh
-rwxr-xr-x 1 root root 1051 Oct 24 16:09 copy_script.sh
```

You can also display the contents of a script file.

```
device# show file copy_script.sh
```

Creating user-defined scripts in the MMVM Linux shell

You can create scripts by using the vi editor in the MMVM Linux shell similar to the following example.

```
[admUser@SLX]# vi create_script.sh
```

For information on accessing the MMVM Linux shell from the SLX-OS device, see Accessing the Linux shell from the SLX-OS CLI on page 66.

After you write the script, make sure that the scripts exist in the `/fabos/users/admin` directory and are executable under the Linux OS.

```
[admUser@SLX]#pwd
/fabos/users/admin
```

Running user-defined scripts from the SLX-OS CLI

With the admin role, you can run the script file directly from SLX-OS CLI. To run a user-defined script, enter the **oscmd** command followed by the name of the script.

NOTE

The **oscmd** command is only visible in the CLI when a user has been configured with the admin role.

```
device# oscmd <my_script>
```

In this example, <my_script> is the name of a user-defined script that either exists in the `/fabos/users/admin` directory and is executable under the Linux OS or downloaded by the **copy** command.

Saving and appending show command output to a file

In the SLX OS, you can save the output of a **show** command. You can also append the output to an existing file.

The **save** and **append** filter options at the SLX-OS CLI are only available for users configured with the admin role.

1. Save the **show** command output to a file.

```
device# show ssh server status | save /tmp/status
```

In this example, the **show ssh server status** output is saved to the /tmp/status file.

2. Display the contents of the file with the Linux **cat** command.

    ```
    device# oscmd cat /tmp/status
    VRF-Name: mgmt-vrf      Status: Enabled
    VRF-Name: default-vrf   Status: Enabled
    ```

3. Append the **show** output to an existing file.

    ```
    device#  show interface ethernet 1/1 | last 5 | append /tmp/status
    ```

    In this example, the **show interface ethernet 1/1** output is appended to the /tmp/status file.

4. Display the contents of the file with the Linux **cat** command.

    ```
    device# oscmd cat /tmp/status
    VRF-Name: mgmt-vrf         Status: Enabled
    VRF-Name: default-vrf      Status: Enabled
    Rate info:
        Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
        Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Time since last interface status change: 01:40:43

    device#
    ```

Logging support for MMVM Linux shell activities

By default, the Brocade device logs the user entering and exiting the MMVM Linux shell, and all user commands executed at the MMVM Linux shell.

MMVM Linux shell user entry and exit logs

The Brocade device uses RASLOG to log entries when the user enters and exits the MMVM Linux shell. If you configure a remote Syslog server, the same logs can also be seen on this server.

On the device, to display the RASLOG entries, use the **show logging raslog** command from privileged EXEC mode.

*   When a user enters the MMVM Linux shell, the **show logging raslog** command displays the following message identified with the SH-1001 ID.

    ```
    device# show logging raslog

    2016/06/25-06:42:54, [SH-1001], 1547, M1 | Active, INFO, SLX,  MMVM Linux shell login information:
    User [admUser]. Login Time : Sat Jun 25 06:42:54 2016
    ```

*   When a user exits the MMVM Linux shell, the **show logging raslog** command displays the following message identified with the SH-1002 ID.

    ```
    device# show logging raslog

    2016/06/25-06:43:59, [SH-1002], 1548, M1 | Active, INFO, SLX, Event: exit, Status: success, Info:
    User [admUser] successfully exited from MMVM Linux shell. Exit Time: Sat Jun 25 06:43:59 2016
    ```

    NOTE
    A message identified with the SH-1003 ID indicates that the user failed to log into the MMVM Linux shell.

MMVM Linux shell command execution logs

Command activities at the MMVM Linux shell are logged locally in the `/var/log/shell_activity.log` file and remotely on a configured remote Syslog server.

When a user executes any command at the MMVM Linux shell, the shell_activity.log file includes messages with the SH-1005 ID similar to the following messages.

```
[admuser@SLX]# tail -f /var/log/shell_activity.log

Jun 25 06:43:28 10.24.12.113 [log value="MMVM Linux shell activity log"] [timestamp value="Sat Jun 25
06:43:28 GMT 2016"] [msgid value="SH-1005"] [severity value="INFO"] [user value="admUser" desc="username"]:
pwd
Jun 25 06:43:29 10.24.12.113 [log value="MMVM Linux shell activity log"] [timestamp value="Sat Jun 25
06:43:29 GMT 2016"] [msgid value="SH-1005"] [severity value="INFO"] [user value="admUser" desc="username"]:
ls
```

> **NOTE**
> The `/var/log/shell_activity.log` file is rotated every thirty minutes if it goes over 2 MB in size. The old version of the file is compressed and a maximum of four rotated files can exist at the same time.

Configuring the remote logging of MMVM Linux shell command activities

By default, the Brocade device logs all the commands executed at the MMVM Linux shell locally in the /var/log/shell_activity.log file and remotely on a configured remote Syslog server.

From SLX-OS CLI, you can perform the following tasks to control the logging of commands executed at the MMVM Linux shell to a remote Syslog server. These tasks will not affect the local logging.

> **NOTE**
> Changes of the **log-shell stop** and **log-shell start** commands are applicable only on new MMVM Linux shell sessions.

1. Disable remote logging.

   ```
   device# log-shell stop
   ```

   Local logging of user activities continues.

2. Restart remote logging,

   ```
   device# log-shell start
   ```

3. Check the remote logging status.

   ```
   device# log-shell status
   ```

   When remote logging is enabled, the following message is displayed.

   ```
   Linux shell activity logging : Enabled
   ```

Python Event-Management and Scripting

Python under Brocade operating systems

The Python interpreter installed with supported Brocade operating systems enables you to access a Python shell or to launch Python scripts. You can also define event handlers that run such scripts automatically upon specified conditions.

> **NOTE**
> SLX-OS is among the Brocade operating systems that support Python.

Python overview

Python is a high-level scripting language that also supports object-oriented programming. If you have previous programming experience, you can quickly learn how to write useful, simple Python scripts.

> **NOTE**
> For Python resources, refer to http://python.org.

Working interactively in the Python shell

Use this procedure to access a Python shell, within which you can use Python commands that call and manipulate Brocade operating system commands.

> **NOTE**
> The Python shell is accessible only to admin-role users.
> Python syntax is case-sensitive.

1. In privileged EXEC mode, enter **python** to access the Python shell.

   ```
   device# python
   ```

   The `device#` prompt changes to a Python prompt:

   ```
   device# python
   Python 3.4.0 (default, Apr 11 2014, 13:05:18)
   [GCC 4.8.2] on linux
   Type "help", "copyright", "credits" or "license" for more information.
   ```

2. To exit the Python shell and return to the Brocade operating system prompt, enter either:

   - `exit()`
   - `Ctrl-D`

3. To run a Brocade operating system command from within the Python shell, enter the `CLI()` command.

```
>>> cmd_show_running_ve = CLI('show running-config interface ve')
!Command: show running-config interface ve
!Time: Mon Aug 22 16:53:13 2016
```

The statement entered above does two things:

- Runs the **show running-config interface ve** command and displays the result.

- Assigns that command to a Python variable named `cmd_show_running_ve`

4. To run a series of Brocade operating system commands from within the Python shell, separate the commands with `\n`.

```
>>> cmd_config_ve = CLI('configure \n interface ve 101-103')
!Command: configure
 interface ve 101-103
!Time: Mon Aug 22 16:53:13 2016
```

> **NOTE**
> There is a difference between running a sequence of Brocade operating system CLI commands in the Python shell rather than in the standard Brocade operating system interface. Whereas in the standard interface the result of a command is persistent, in the Python shell each `CLI( )` statement is independent of any preceding ones.

In the following example, the lines beginning with **#** are added for explanation.

```
device# python
Python 3.4.0 (default, Apr 11 2014, 13:05:18)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> cmd_show_running_ve = CLI('show running-config interface ve')
!Command: show running-config interface ve
!Time: Mon Aug 22 16:53:13 2016

% No entries found.
# The SLX-OS show running-config interface ve command is run,
# and that command is assigned to the Python variable cmd_show_running_ve.

>>> cmd_config_ve = CLI('configure \n interface ve 101-103')
# A series of three commands are run and assigned to the Python variable cmd_config_ve.
!Command: configure
 interface ve 101-103
!Time: Mon Aug 22 16:53:13 2016


>>> cmd_show_running_ve.rerun()
# The rerun() function appended to cmd_show_running_ve gives the following output:
!Command: show running-config interface ve
!Time: Mon Aug 22 16:53:13 2016

interface Ve 101
 shutdown
!
interface Ve 102
 shutdown
!
interface Ve 103
 shutdown
!
!
```

Python scripts

Python scripts enable you to manipulate and launch Brocade operating system commands, taking advantage of the power and flexibility of Python. Such scripts also support event handling.

The topics in this section guide you through the process of writing and testing Python scripts, copying them to supported devices, and running them with the **python** command from the command line.

Guidelines for writing Python scripts

The guidelines for writing Python scripts to run under SLX-OS are as follows:

- Although previous experience programming in Python is helpful, experience in other high-level languages is enough to get you started with simple Python scripts.
- The Python developer either needs experience with SLX-OS CLI or access to a resource with such experience.
- To help decide which editor or integrated development environment (IDE) to use, refer to http://www.python.org.
- Make sure that the appropriate version of the Python interpreter is installed on your development computer. For the current version of SLX-OS, install Python 3.4.0.
- Make sure that in the *Brocade SLX-OS Command Reference* you are familiar with the **python** and the **CLI( )** topics.
- One of the first statements in the script should be `from CLI import CLI`. This enables the `CLI( )` command, by which Python can interact with SLX-OS.
- Write the Python script and save it, with a `.py` suffix. Valid filenames range from 4 through 32 characters (including the suffix). The first character must be alphabetic.

  NOTE
  For additional sample scripts, refer to Python scripts and run-logs on page 123.

Testing Python-script statements

While developing a Python script, you can test Brocade operating system calls by entering them in the device Python command shell. After the script is stable, you copy it to the device and then test it further by running it from the command line.

1. In privileged EXEC mode, enter the **python** command to access the Python shell.

   ```
   device# python
   Python 3.4.0 (default, Apr 11 2014, 13:05:18)
   [GCC 4.8.2] on linux
   Type "help", "copyright", "credits" or "license" for more information.
   >>>
   ```

   Note that the `device#` prompt changed to a >>> Python prompt:

2. Enter the script statements one at a time, verifying that they run as expected.

   ```
   >>> cmd_config_ve = CLI('configure \n interface ve 101-103')
   !Command: configure
    interface ve 101-103
   !Time: Mon Aug 22 16:53:13 2016
   ```

3. Make corrections as needed.

Copying Python files to the device

After writing and testing a Python script file, use one of these topics to copy it to device flash memory.

Copying a file from a USB device

Use this topic to copy a file from a USB stick to a Brocade device .

ATTENTION
The only supported USB device for this task is the Brocade USB stick shipped with the device.

1. Copy the Python script file to the USB stick.
2. Insert the USB stick into the device USB port and enter the **usb on** command.
3. In privileged EXEC mode, enter the **copy** command to copy the Python file from the USB stick to the device flash memory.

```
device# copy usb://pythscript1.py flash://pythscript1.py
```

4. To display a list of files in the device flash memory, enter the **dir** command.

```
device# dir
total 32
drwxr-xr-x 2  251 1011  4096  Aug 26 08:44 .
drwxr-xr-x 3  251 1011  4096  Jul 20 07:50 ..
-rw-r--r-- 1 root root  1051  Mar 24 16:09 create_po.py
-rw-r--r-- 1 root sys    695  Aug 23 21:18 defaultconfig.cluster
-rw-r--r-- 1 root root   410  Aug 26 04:06 defaultconfig.standalone
-rw-r--r-- 1 root root 10042  Aug  4 00:01 ospfnsr1.cfg
-rw-r--r-- 1 root root   410  Aug 26 01:28 startup-config

16908197888 bytes total (10226114560  bytes free)
```

5. To display the contents of a Python file copied into the device flash memory, enter the **show file** command.

```
device# show file pythscript1.py
```

Downloading a file from a network

Use this topic to download a file from a network location to the Brocade device.

1. Make sure that the Python script file is uploaded to an accessible network location.
2. In privileged EXEC mode, enter the **copy** command to copy the Python file from the network location to the device flash memory.

```
device# copy ftp://MyUserID:MyPassword@10.10.10.10//pythscript1.py flash://pythscript1.py
```

For other file-transfer options, refer to the *Brocade SLX-OS Command Reference* **copy** topic.

3. To display a list of files in the device flash memory, enter the **dir** command.

```
device# dir
total 32
drwxr-xr-x 2  251 1011  4096  Aug 26 08:44 .
drwxr-xr-x 3  251 1011  4096  Jul 20 07:50 ..
-rw-r--r-- 1 root root  1051  Mar 24 16:09 create_po.py
-rw-r--r-- 1 root sys    695  Aug 23 21:18 defaultconfig.cluster
-rw-r--r-- 1 root root   410  Aug 26 04:06 defaultconfig.standalone
-rw-r--r-- 1 root root 10042  Aug  4 00:01 ospfnsr1.cfg
-rw-r--r-- 1 root root   410  Aug 26 01:28 startup-config

16908197888 bytes total (10226114560  bytes free)
```

4. To display the contents of a Python file copied into the device flash memory, enter the **show file** command.

```
device# show file pythscript1.py
```

Running Python scripts from the command line

The facility to run Python scripts from the Brocade operating system command line enables you to execute complex and repetitious tasks with accuracy and efficiency. This facility also enables you to validate scripts intended for event management.

> **CAUTION**
> **Make sure that you test Python scripts according to standard quality assurance practices before deploying them.**

In privileged EXEC mode, enter the **python** command, specifying the Python script file that you want to run.

```
device# python create_po.py
```

After the script runs, the SLX-OS prompt displays.

> **NOTE**
> The create_po.py script is discussed in "Script for assigning interfaces to port channels (create_po.py)."

Python scripts and run-logs

This section contains sample SLX-OS Python scripts and run-logs.

Script for assigning interfaces to port channels (create_po.py)

The `create_po.py` script is an example for automating common configuration tasks.

Script (create_po.py)

This script runs the relevant **show running-config** commands before and after the following actions:

- VLAN configuration
- Port-channel configuration
- Adding interfaces to port channels

> **NOTE**
> Lines beginning with # are annotations.

```
#Required in all scripts for SLX-OS:
from CLI import CLI

slots = [1, 2]
interfaces = [40, 41, 42, 43]
port_channel = 10
vlan_range = "101-105"

# Runs show running-config int vlan before the configuration, and assigns this SLX-OS
# command to a Python variable named cmd_show_running_vlans.
cmd_show_running_vlans = CLI("show running-config vlan")

# Configures VLANs.  {} is a placeholder for the format (arg1, arg2, ... argN) variables:
cmd_configure_vlans = CLI("config \n vlan {}".format(vlan_range))

# Reruns show running-config int vlan, following definition of new VLANs:
```

```
cmd_show_running_vlans.rerun()

# Configures port channel (vLAG):
cmd_show_running_port_channels = CLI("show running-config int po")
cmd_configure_port_channel = CLI("config \n int po {} \n switchport \n switchport mode trunk \n switchport
trunk allowed vlan add {} \n switchport trunk tag native-vlan ; no shut".format(port_channel, vlan_range))

# Reruns show running-config int po, following configuration changes:
cmd_show_running_port_channels.rerun()

# Add interfaces to port channel (vLAG)
for slot in slots:
    for interface in interfaces:
        cmd_configure_interface = CLI("config \n int eth {}/{} \n channel-group {} mode active type
standard \n no shut".format(slot, interface, port_channel))
        cmd_show_running_int_eth = CLI("show running-config int eth {}/{}".format (slot, interface))

# Runs show running-config:
cmd_show_running = CLI("show running-config")
```

Run-log (create_po.py)

A log upon running `create_po.py` was as follows:

```
device# python create_po.py
!Command: show running-config vlan
!Time: Mon Aug 22 18:33:03 2016

vlan 1
!
vlan dot1q tag native

!Command: config
 vlan 101-105
!Time: Mon Aug 22 18:33:03 2016


!Command: show running-config vlan
!Time: Mon Aug 22 18:33:03 2016

vlan 1
!
vlan 101
!
vlan 102
!
vlan 103
!
vlan 104
!
vlan 105
!
vlan dot1q tag native

!Command: show running-config int po
!Time: Mon Aug 22 18:33:03 2016

interface Port-channel 1
 description Insight port-channel on MM1
 shutdown
!
interface Port-channel 2
 description Insight port-channel on MM2
 shutdown
!

!Command: config
 int po 10
 switchport
```

```
 switchport mode trunk
 switchport trunk allowed vlan add 101-105
 switchport trunk tag native-vlan ; no shut
!Time: Mon Aug 22 18:33:03 2016


!Command: show running-config int po
!Time: Mon Aug 22 18:33:04 2016

interface Port-channel 1
 description Insight port-channel on MM1
 shutdown
!
interface Port-channel 2
 description Insight port-channel on MM2
 shutdown
!
interface Port-channel 10
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 101-105
 switchport trunk tag native-vlan
 no shutdown
!

!Command: config
 int eth 1/40
 channel-group 10 mode active type standard
 no shut
!Time: Mon Aug 22 18:33:04 2016


!Command: show running-config int eth 1/40
!Time: Mon Aug 22 18:33:04 2016

interface Ethernet 1/40
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!

!Command: config
 int eth 1/41
 channel-group 10 mode active type standard
 no shut
!Time: Mon Aug 22 18:33:04 2016


!Command: show running-config int eth 1/41
!Time: Mon Aug 22 18:33:05 2016

interface Ethernet 1/41
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!

!Command: config
 int eth 1/42
 channel-group 10 mode active type standard
 no shut
!Time: Mon Aug 22 18:33:05 2016


!Command: show running-config int eth 1/42
!Time: Mon Aug 22 18:33:05 2016

interface Ethernet 1/42
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
```

```
!Command: config
 int eth 1/43
 channel-group 10 mode active type standard
 no shut
!Time: Mon Aug 22 18:33:05 2016


!Command: show running-config int eth 1/43
!Time: Mon Aug 22 18:33:05 2016

interface Ethernet 1/43
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
<output truncated>
```

Script illustrating the .get_output function (get_output.py)

The .get_output function returns—as a list—the output of the SLX-OS CLI commands assigned to a Python object.

Script (get_output.py)

This script displays a list of firmware version per slot.

> **NOTE**
> Lines beginning with # are annotations.

```
#Required in all scripts for SLX:
from CLI import CLI
# Import the Python Regular Expressions (re) module:
import re
# Create Python objects:
slot_firmware = {}

cmd_show_ver = CLI("show ver", False)
# Using .get_output(), assign the result of show ver to a Python object named output:
output = cmd_show_ver.get_output()
for line in output:
    found = re.search(r'^(\S+)\s+(\S+)\s+(\S+)\s+ACTIVE.*$', line, re.M)
    if found:
        slot_firmware[found.group(1)] = found.group(3)

print("SLOT_FIRMWARE:\n")
for key in slot_firmware:
    print("\t", key, "\t=> ", slot_firmware[key])
```

Python event-management

Python event management enables you to specify a Python script that runs automatically upon specified conditions.

You specify which RASlog message triggers the script.

Configuring an event-handler profile

Use this procedure to create an event-handler profile, define one or more triggers for it, and specify a Python script that runs upon one or more trigger events.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **event-handler** command.

   ```
   device(config)# event-handler eventHandler1
   ```

3. For each trigger that you need for a profile, enter the **trigger** command.

   ```
   device(config-event-handler-eventHandler1)# trigger 1 raslog LOG-1001
   ```

   The trigger event is RASlog message #LOG-1001.

4. Enter the **action python-script** command to specify a Python script that runs when the event-handler is triggered.

   ```
   device(config-event-handler-eventHandler1)# action python-script example.py
   ```

   ⚠️ CAUTION
   Make sure that you test Python scripts according to standard quality assurance practices before deploying them.

The following example includes all of the steps.

```
device# configure terminal
device(config)# event-handler eventHandler1
device(config-event-handler-eventHandler1)# trigger 1 raslog LOG-1001
device(config-event-handler-eventHandler1)# action python-script example.py
```

Activating an event-handler

Use this procedure to activate one or more event-handlers on the device. If a trigger specified in the event-handler profile occurs, a designated Python script runs.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **event-handler activate** command, specifying the event handler that you are activating.

   ```
   device(config)# event-handler activate eventHandler1
   ```

3. To activate an additional event handler, enter the **event-handler activate** command, specifying the additional event handler.

   ```
   device(config-activate-eventHandler1)# event-handler activate eventHandler2
   ```

4. To de-activate an event handler, enter the **no event-handler activate** command

   ```
   device(config-activate-eventHandler2)# no event-handler activate eventHandler2
   ```

The following example includes all of the activation steps.

```
device# configure terminal
device(config)# event-handler activate eventHandler1
device(config-activate-eventHandler1)# event-handler activate eventHandler2
```

Configuring event-handler options

After you activate an event handler, you can configure various options. For example, you can specify if the event-handler script runs more than once and how multiple triggers are handled.

1. Activate an event handler, as described in Configuring an event-handler profile on page 127:

   ```
   device# configure terminal
   device(config)# event-handler activate eventHandler1
   ```

2. To specify a delay from when a trigger is received until execution of the event-handler action, enter the **delay** command.

   ```
   device(config-activate-eventHandler1)# delay 60
   ```

   The above example specifies a delay of 60 seconds.

3. To specify multiple iterations of the action when a trigger is received:

   a) Enter the **iterations** command.

   b) To specify an interval between iterations, enter the **interval** command.

   ```
   device(config-event-handler-eventHandler1)# iterations 3
   device(config-activate-eventHandler1)# interval 30
   ```

   The above example sets the number of iterations to 3 and specifies an interval of 30 seconds between each iteration.

4. To limit action-recurrence upon multiple trigger-events, enter one of the following commands:

   - **trigger-mode only-once**—for the duration of a device configuration, the event-handler action is launched only once.

     ```
     device(config-activate-eventHandler1)# trigger-mode only-once
     ```

   - **trigger-mode on-first-instance**—as long as the device is running, the event-handler action is launched only once. Following a device restart, the event-handler action can be triggered again.

     ```
     device(config-activate-eventHandler1)# trigger-mode on-first-instance
     ```

5. If multiple triggers are defined, to specify that the action run only if all of the triggers occur, enter the **trigger-function AND time-window** command.

   ```
   device(config-activate-eventHandler1)# trigger-function AND time-window 120
   ```

   The above example specifies that the action run only if all triggers occur within 120 seconds.

Troubleshooting event-management

Use these topics to troubleshoot issues that arise during implementation of Python event-management.

Aborting an event-handler action

If needed, abort a Python script launched by an event-handler action.

1. In privileged EXEC mode, enter the **clear event-handler action** command.

   ```
   device# clear event-handler actionn eh1
   This operation will abort an event handler action that is currently running and may leave the device
   in an inconsistent state.  Do you want to continue? [y/n]:y
   ```

2. To confirm aborting the action, type `y`.

```
Operation completed successfully.
```

The Python script launched by the event-handler action is aborted.

Event-management show commands

There are several **show** commands that display event-management information. They are documented in the *Brocade SLX-OS Command Reference*, and listed here with descriptions.

TABLE 7 Event-management show commands in the *Brocade SLX-OS Command Reference*

Command	Description
show event-handler activations	Displays operational data of activated event-handlers.
show running-config event-handler	Displays details of event-handler profiles defined on the device. You can display the results by Python-script action or trigger ID.