Cisco Application eXtensions Platform User Guide

Revised: 2/8/08, OL-14815-01

This guide describes the commands and tasks for installing and configuring the Cisco Application eXtensions Platform (AXP) on Cisco's Integrated Service Routers. To create third party applications for the Cisco AXP, see the *Cisco AXP Developer Guide*.

Use this guide to:

- Set up the router and the service module.
- Configure the service module interface.
- Configure the application environment on the service module.
- Troubleshoot the application environment.

This guide consists of the following sections:

Overview

The Cisco Integrated Services Router (Cisco ISR) is an integrated system within a single chassis. The Cisco ISR ties together and runs multiple value-added services such as voice, layer 2 switching, security and application acceleration. In addition, integrated services can be hosted within Cisco OS software or decoupled and hosted on modular application service modules.

The Cisco ISR allows for blade hardware plug-in network modules. These application service modules enhance the functionality, intelligence and flexibility of the router. The Cisco Application eXtensions Platform (Cisco AXP) provides the tools required by third party developers to integrate their applications on Cisco ISRs.

Cisco AXP allows third parties such as system integrators, managed service providers, and large enterprise customers to extend the functionality of Cisco ISRs by providing their own value-added integrated services. On the application service module, Cisco AXP hosts applications in a separate runtime environment with dedicated resources. In addition, Cisco AXP provides Application Programming Interfaces (APIs) so that functions such as packet analysis, event notification, and network management can be utilized by hosted applications.

Cisco AXP consists of the facilities and frameworks to host applications, and service APIs for integrating applications into the network.

Cisco AXP provides the following features:

- Ability to modify the Cisco IOS software configuration and obtain the status of Cisco IOS software features via the provided API.
- Embedded Linux environment supporting the execution of applications written in the following programming languages: Java, C (native), Perl (interpreted), Python (interpreted), and Bash (interpreted). Native and interpreted applications written in other programming languages can be integrated by the application vendor if the vendor uses additional support libraries and interpreters.
- Integration of virtual devices.

  The Cisco IOS auxiliary serial port can be virtualized, appearing as a local device in the Cisco AXP OS. The application controls external peripherals attached to the auxiliary serial port of the router without special knowledge of where the device is located.
- Predictable and constant set of application resources.

  These resources (including CPU, memory, and disk) are segmented, which ensures that the application and router features work independently, and without interference.
- Protection of the router and applications from rogue applications.

  If an application crashes, this incident does not affect the router or other applications, because of the installed application being placed in its own virtual instance.
- Protection against running unauthorized software.

  Only Cisco certified parties can install software on Cisco AXP.
- Robust debugging and troubleshooting facilities.
- Support of event notification.

  An application can receive the status of the Cisco ISR and take appropriate action.

Note For a more detailed overview of Cisco AXP, see the *Cisco AXP Developer Guide*.

Initial System Setup

To set up your router and Cisco AXP service module:

1. Verify that the service module SKU matches the Cisco ISR and install the router and service module. See the "Hardware Requirements" section on page 4.
2. Download the applicable Cisco IOS software image and Cisco AXP files and configure the router. See the "Software Requirements" section on page 5.
3. Review the Cisco AXP command environment. See the "Entering and Exiting the Command Environment" section on page 6.
4. Configure the Cisco AXP service module interface. See the "Configuring the Cisco AXP Service Module Interface" section on page 8.
5. Install core and add-on packages, upgrade software versions, and use the software helper image, see the "Installing and Upgrading Software" section on page 13.
6. Configure secure shell (SSH) access. See the "Secure Shell Access to the Service Module" section on page 17.
7. Configure the syslog server. See the "Configuring the Syslog Server" section on page 20 and "Verifying the Syslog Server" section on page 25.

Hardware Requirements

Verify your service module and router compatibility, see Table 1.

Table 1 Cisco AXP Service Module and Router Compatibility

Cisco AXP Service Module SKU	Supported Routers	Processor/Memory	Hard Disk	Compact Flash
AIM-APPRE-102-K9	Cisco 1841, Cisco 2801, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3825, Cisco 3845.	300 MHz/256 MB	—	1 GB
NME-APPRE-302-K9	Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3825, Cisco 3845.	1.0 GHz/512 MB	80 GB	—
NME-APPRE-522-K9	Cisco 3825, Cisco 3845.	1.4 GHz/2.0 GB	160 GB	—

CPU Resources

Cisco AXP provides a predictable and constant set of resources such as CPU, memory, and disk space, which are segmented to allow an application to run on the Cisco AXP service module without affecting the performance of other features of the router.

You can specify CPU, memory, and disk space limit (specified in MB), using the CLI with administrator privileges.

For information on the CPU index for the Cisco AXP service modules, see the Dedicated Application Resources section in the *Cisco AXP Developer Guide*.

Installation

For information on router and service module installation, see the relevant hardware installation documentation at:
http://www.cisco.com/en/US/products/hw/modules/ps2797/prod_installation_guides_list.html.

Software Requirements

Download the applicable Cisco IOS software image and Cisco AXP package files.

Cisco IOS Software Release

- 12.4(15)T: IP-based crypto image including the following image packs:
  - IP-Base
  - IP-Voice
  - Adv-Security
  - Adv-Enterprise

  For the event trigger API the following image packs are supported:
  - IP-Voice
  - Adv-Security
  - Adv-Enterprise

Download the image from:

http://www.cisco.com/kobayashi/sw-center/

Cisco AXP Software

Prerequisites

- IP address or name of the FTP server that will store the Cisco AXP package file.
- Verify that the FTP server is accessible

Summary Steps

1. Download the Cisco AXP files.

   For further information on files, see the *Cisco AXP Release Notes.*

2. Copy the files to the FTP server.

   To install the Cisco AXP files, see the "Installing and Upgrading Software" section on page 13.

Application Software Versions

Each application has a version number, which is used by the Installer when resolving dependencies between subsystems. The system only considers the first two digits of the version number for checking dependencies.

The first digit is treated as a major version number and the second is treated as a minor version number. For example:

> Version 1.2.3.4 has a major number of 1 and minor number of 2.
>
> Version 5.6 has a major number of 5 and minor number of 6.
>
> Version 1.2.3 is considered the same as Version 1.2 or Version 1.2.3.4 because both the major and minor numbers match.

Entering and Exiting the Command Environment

This section includes the procedures listed below for entering and exiting the command environment, where Cisco AXP software configuration commands are executed.

- Entering the Command Environment, page 6
- Exiting the Command Environment, page 7

EXEC and Configuration Modes

The Cisco AXP software command modes, EXEC and configuration, operate similarly to the EXEC and configuration modes for Cisco IOS software CLI commands.

Entering the Command Environment

To enter the command environment after the Cisco AXP software is installed and active, perform the following steps.

Prerequisites

The following information is required to enter the command environment:

- IP address of the Cisco ISR router that contains the Cisco AXP service module
- Username and password to log in to the router
- Slot number of the module

SUMMARY STEPS

1. Open a Telnet session.
2. **telnet** *ip-address*
3. Enter the user ID and password of the router.
4. **service-module service-engine** *slot*/*port* **session**
5. (Optional) **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Open a Telnet session.	Use a DOS window, a secure shell, or a software emulation tool such as Reflection.

Step 2	`telnet` *`ip-address`*
	Example:
	`C:\>telnet 172.16.231.195`
		Specifies the IP address of the router.

Step 3	`Username:`
	`Password:`
		Enter your username and password for the router.

Step 4	`service-module integrated-service-engine` *`slot/port`* `session`
	Example:
	`Router# service-module integrated-service-engine 1/0 session`
		Enters the Cisco AXP software command environment using the module located in *slot* and *port*. The prompt changes to "se" with the IP address of the service module.
		If the message,
		`"Trying ip-address slot/port ...`
		`Connection refused by remote host"`
		appears, enter the command
		`service-module integrated-service-engine` *`slot/port`* `session clear`
		and try Step 4 again.

Exiting the Command Environment

To exit the Cisco AXP software command environment and return to the router command environment, perform the following steps.

SUMMARY STEPS

Return to the Cisco AXP software EXEC mode.

1. **exit**.
2. **exit**.

DETAILED STEPS

Command or Action	Purpose
Return to the Cisco AXP software EXEC mode.	

	Command or Action	Purpose
Step 1	`exit`	

Step 2	`exit`
	Example:
	`se-10-0-0-0# exit`
	`se-10-0-0-0> exit`
	`router-prompt#`
		Returns to the router command environment.

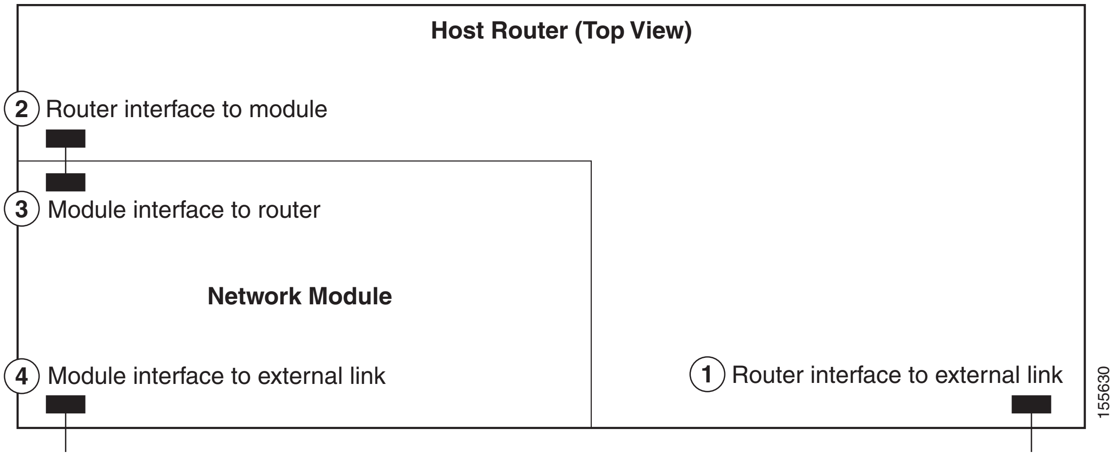
Configuring the Cisco AXP Service Module Interface

The host router and service module use two main interfaces for internal and external communication (see Figure 1).

- Internal Interface (eth0)
  - This interface is internal, between the router and the service module. Use this connection to exchange traffic between the network interface and the router.
  - For example, the console connection to the service module is connected through this interface.
  - On the router side, this interface is described as service interface engine x/0 (where x is the service module slot in which the service module is inserted).
  - On the service module side (Linux), this internal interface is designated as eth0.
- External Interface (eth1)
  - This an external interface, and it varies between a Fast (100 Mb/s) or Gigabit (1000 Mb/s) ethernet and is available through an RJ-45 connector.
  - On the service module side, this external interface is designated as eth1.

Note The external interface is not visible from the router side and can only be configured and used from the service module.

Figure 1 Router and Service Module Interfaces


Host Router (Top View)
2 Router interface to module
3 Module interface to router
Network Module
4 Module interface to external link
1 Router interface to external link
155630

	On This Hardware Interface...	Configure These Settings...	Using This Configuration Interface
1	Router interface to external link (FastEthernet *slot*/0)	Standard router settings	Cisco IOS software CLI on the router.
2	Router interface to module (Integrated-service-engine *slot*/0)	Module's IP address and default gateway router	
3	Module interface to router (eth0)	All other module and Cisco AXP software application settings	Cisco AXP software GUI, or SSH interface.
4	Module interface to external link (eth1)	Support for data requests and transfers from outside sources	

Configuration Tasks

Steps 1 to 3 open the host-router CLI and access the router's interface to the module.
Steps 4 to 9 configure the interface.

SUMMARY STEPS

From the Router CLI

1. **enable**
2. **configure terminal**
3. **interface integrated-service-engine** *slot*/**0**
4. **ip address** *router-side-ip-address subnet-mask*

   or

   ip unnumbered *type number*
5. **service-module ip address** *module-side-ip-address subnet-mask*
6. **service-module ip default-gateway** *gateway-ip-address*
7. **end**

8. **copy running-config startup-config**
9. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
	From the Host-Router CLI	
Step 1	**enable** **Example:** Router> enable	Enters privileged EXEC mode on the host router. Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode on the host router.
Step 3	**interface integrated-service-engine** *slot***/0** **Example:** Router(config)# **interface integrated-service-engine 1/0**	Enters interface configuration mode for the slot and port where the service module resides. • *slot*—Specifies the service module slot.
Step 4	**ip address** *router-side-ip-address subnet-mask* or **ip unnumbered** *type number* **Example:** Router(config-if)# **ip address 10.0.0.20 255.255.255.0** or Router(config-if)# **ip unnumbered GigabitEthernet 0/0**	The **ip address** command specifies the IP address on the router side. • *router-side-ip-address subnet-mask*—IP address and subnet mask for the interface. The **ip unnumbered** command enables IP processing on an interface without assigning an explicit IP address to the interface. • *type*—Type of interface on which the router has an assigned IP address. The interface cannot be another unnumbered interface. • *number*— Number of the interface on which the router has an assigned IP address. The interface cannot be another unnumbered interface. For more information on the IP unnumbered command, see: *Understanding and Configuring the IP Unnumbered Command*.
Step 5	**service-module ip address** *module-side-ip-address subnet-mask* **Example:** Router(config-if)# **service-module ip address 172.0.0.20 255.255.255.0**	Specifies the IP address for the module interface to the router. • *module-side-ip-address*—IP address for the interface • *subnet-mask*—Subnet mask to append to the IP address; must be in the same subnet as the host router

	Command or Action	Purpose

Step 6	**`service-module ip default-gateway`** *`gateway-ip-address`*
	Example:
	`Router(config-if)# service-module ip default-gateway 10.0.0.40`
		Specifies the IP address for the default gateway router for the module. The argument is as follows:
		• *gateway-ip-address*—IP address for the gateway router

Step 7	**`end`**
	Example:
	`Router(config-if)# exit`
		Returns to global configuration mode on the host router.

Step 8	**`copy running-config startup-config`**
	Example:
	`Router# copy running-config startup-config`
		Saves the router's new running configuration.

Step 9	**`show running-config`**
	Example:
	`Router# show running-config`
		Displays the router's running configuration, so that you can verify address configurations.

Opening and Closing a Service Module Session

To open and close a session on the service module, perform the following steps.

Note

- Before you install your application software, opening a session brings up the bootloader. After you install the software, opening a session brings up the application.
- You can conduct only one session at a time.
- Steps 1 to 3: open the host-router CLI and access the module. Steps 4 to 5: configure the module. Step 6: return to the host-router CLI.

SUMMARY STEPS

From the Host-Router CLI

1. **enable**
2. **service-module integrated-service-engine** *slot***/0 status**
3. **service-module integrated-service-engine** *slot***/0 session**

From the Service-Module Interface

4. **Enter configuration commands**
5. **Control-Shift-6 x**

From the Host-Router CLI

6. **service-module integrated-service-engine** *slot***/0 session clear**

DETAILED STEPS

	Command or Action	Purpose
	From the Host-Router CLI	

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enters privileged EXEC mode on the host router. Enter your password if prompted.

Step 2	**`service-module integrated-service-engine`** *`slot`***`/0 status`**
	Example:
	`Router# service-module integrated-service-engine 1/0 status`
		Displays the status of the specified module, so that you can ensure that the module is running (that is, in the steady state).

Step 3	**`service-module integrated-service-engine`** *`slot`***`/0 session`**
	Example:
	`Router# service-module integrated-service-engine 1/0 session`
	`Trying 10.10.10.1, 2065 ... Open`
		Begins a service-module session on the specified module. Do one of the following:
		• To interrupt the autoboot sequence and access the bootloader, quickly type ***.
		• To start a configuration session, press **Enter**.

	From the Service-Module Interface	

Step 4	**.Example (Software installation):**
	`SE-Module>` **`software install add url`** *`Url`*
		Enter configuration commands on the module as needed.
		• Bootloader command choices include **boot**, **config**, **exit**, **help**, **ping**, **reboot**, **show**, and **verify**.
		– To configure the bootloader after setting up the router, see the "Configuring the Bootloader" section on page 98.
		• To install the application software:
		Use the **software install add url** command. See the "Installing and Upgrading Software" section on page 13 for details.
		• Configuration command choices are similar to those that are available on the router. Access global configuration mode by using the **configure terminal** command.
		• Enter configuration commands.
		• Exit global configuration mode with the **exit** command and save your new configuration with the **copy running-config startup-config** command. Notice that you do not use the **enable** command and the prompt does not change from **>**.

	Command or Action	Purpose
Step 5	Press **Control-Shift-6 x**.	Suspends the service-module session and returns to the router CLI. Alternatively, type the **exit** command to return to the router CLI. **Note** The service-module session stays up until you clear it in the next step. While it remains up, you can return to it from the router CLI by pressing **Enter**.
	From the Host-Router CLI	

Step 6	**`service-module integrated-service-engine`** *`slot`***`/0 session clear`**
	Example:
	`Router# service-module integrated-service-engine 1/0 session clear`
		Clears the service-module session for the specified module. When prompted to confirm this command, press **Enter**.

Installing and Upgrading Software

This section contains the following tasks:

- Installing Software, page 13
- Installing Software including FTP Server Configuration, page 14
- Add-on Installation, page 15
- Upgrading Software, page 15
- Installing Software using a Helper Image, page 16

This section provides configuration tasks to install core Cisco AXP packages and the application software add-on packages.

Note See the *Cisco AXP Developer Guide* for creating and packaging third party applications.

Installing Software

To install the Cisco AXP package that was downloaded on the FTP server, perform the following steps.

Note Using the **software install clean** command clears the startup configuration to the factory default setting. All previous configuration settings are erased.

Prerequisites

Before you install the package, you need:

- FTP server user ID
- FTP server password

SUMMARY STEPS

1. **software install clean** {*package-filename* | **url** *ftp://ftp-server-ip-address/package-filename*} **username** *username* **password** *password*
2. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`software install clean url {package-filename	url ftp://ftp-server-ip-address/package-filename} username username password password`
		`Example:`
		`SE-module> software install clean url ftp://10.10.1.5/.../.../packagename.pkg username johndoe password johndoe123`
			Installs the Cisco AXP package file that was downloaded on the FTP server.
			ftp url/— FTP server address
			package file-name— Cisco AXP package filename
			username—FTP server username
			password—FTP server password

Step 2	`exit`	Exits EXEC mode.

Installing Software including FTP Server Configuration

To configure the FTP server and install the Cisco AXP package, perform the following steps.

Prerequisites

Before you install the package, you need:

- FTP server user ID
- FTP server password

SUMMARY STEPS

1. **configure terminal**
2. **software download server url** *ftp-url[/ dir]* **username** *username* **password** *password*
3. **end**
4. **copy running-config startup-config**
5. **software install clean** *package file-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**software download server url** *ftp-url[/dir]* **username** *username* **password** *password* Example: `SE-Module(config)> software download server url ftp://10.10.1.5/.../.../packagename.pkg` **username** `johndoe` **password** `johndoe123`	Configures the FTP server address. *ftp url*— FTP server address */dir*— (optional) FTP directory on the server *username*—FTP server username *password*—FTP server password
Step 3	**end**	Exits configuration mode.
Step 4	**copy running-config startup-config**	Saves configuration for the download server.
Step 5	**software install clean url** *package file-name* `SE-module>` **software install clean** `packagename.pkg`	Installs the Cisco AXP package file that was downloaded on the FTP server. *package file-name*— Cisco AXP package filename
Step 6	**end**	Exits EXEC mode.

Add-on Installation

To install add-on software on the service module in Cisco AXP EXEC mode, perform the following steps.

SUMMARY STEPS

1. Open a Service Module Session. See the "Opening and Closing a Service Module Session" section on page 11.
2. From the Service Module Interface: **software install add url** *url*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**software install add url** *url* Example: `SE-Module>` **software install add url** `ftp://10.10.1.5/pkgname.1.0.pkg`	Installs the application software from the specified *url*. *url*—location of the application service software.
Step 2	**exit**	Exits EXEC mode.

Upgrading Software

When upgrading software, the application must be packaged with the packaging tool and using particular values for arguments such as *uuid*, *name*, and *version*. See "Packaging Tool" in the *Cisco AXP Developer Guide*.

To upgrade the software on the service module in Cisco AXP EXEC mode, perform the following steps.

Note Using the **software install upgrade** command restores saved configurations.

SUMMARY STEPS

1. **software install upgrade** { *package-filename*.**pkg** | **url ftp**://*ftp-server-ip-address*/*packageName*.**pkg**}
2. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`software install upgrade { package-filename.pkg	url ftp://ftp-server-ip-addr/package-fileame.pkg}`
		`Example:`
		`SE-Module> software install upgrade url ftp://10.10.1.5/pkgname.1.0.pkg`
			Installs the upgraded software version.

Step 2	`exit`	Exits EXEC mode.

Installing Software using a Helper Image

If the service module does not boot up with the regular image you can install software using a helper image. To use the boot helper to reboot with a helper image, perform the following steps.

Note

- Configure the router before setting up the bootloader. The service module will not connect to the external network if the router is not configured correctly. If you have not configured the bootloader, see the "Configuring the Bootloader" section on page 98.
- Using the helper image for installation clears the startup configuration to the factory default settings. All previous configuration settings are erased.

Step 1 Enter the following commands from the router CLI:

a. **service-module Service-Engine 1/0 reset** (wait about 10 seconds after using this command).

b. **service-module Service-Engine 1/0 session** (repeat this command if the first try fails).

Step 2 Wait for the following prompt: Please enter *** to change boot configuration.

a. Enter "***" to drop the service module into the bootloader.

Step 3 Enter the **config** command to configure the bootloader:

SE- boot-loader> **config**

a. Enter these parameters:

IP address—Service module IP address as configured on the host router

Subnet mask—Service module subnet mask as configured on the host router

TFTP server—IP address of the TFTP server with the helper image

Gateway—Gateway address as configured in the host router

Default Helper-file—Filename of the helper image

Ethernet interface—**Internal**

Note The internal interface is facing the router. The external interface may or may not be present on the service module.

External interface media—**copper**

Default Boot—**disk**

Default bootloader—**secondary**

Note Always use the secondary bootloader; the primary bootloader is only for backup.

Note Before entering the boot helper (in step 4), do the following:

- Check that the IP address you entered in step 3 is the IP address of the service module as configured on the host router.
- Check that the TFTP server you entered in step 3 is reachable from the service module.

Step 4 To enter the boot helper type: **boot helper**

The helper image starts and the following text appears:

```
Welcome to Cisco Systems Service Engine Helper Software
Please select from the following
1       Install software
2       Reload module
3       Disk cleanup
(Type '?' at any time for help)
Select 1 Install software:
```

Step 5 Enter the following parameters:

Package Name—Cisco AXP package name

Server URL—FTP server location for the Cisco AXP package

Username—FTP username

Password—FTP password

Step 6 Enter **y** (yes) to clear disk contents.

After installation, the blade reboots with the new image.

Secure Shell Access to the Service Module

For direct secure shell (SSH) access to the Cisco AXP CLI, a user must first session into the service module and configure it.

This initial configuration gives users direct SSH access to the Cisco AXP CLI and also allows them to perform remote configurations without constantly accessing the router and sessioning into the service module.

Cisco AXP provides secure shell (SSH) access to the CLI through a default user that acts like a system administrator. The password for the default system administrator must be configured by the user through the CLI, before SSH access to the Cisco AXP CLI. This password must be at least five characters.

If service password encryption is enabled, the system encrypts the clear text password that is entered and saves it in the encrypted format in the configuration file. If the user prefers a clear text password, service password-encryption is disabled by entering the **no** form of the command.

This command directs the system to encrypt the password using a system provided two-character salt string. With this service enabled, a clear text password string is encrypted and displayed as a level 7 password string.

Cisco AXP also provides commands for SSH tunneling. These commands are documented in the "SSH Access to a Virtual Instance" section on page 63.

This section contains the following tasks:

- Configuring Password Protection, page 18
- Configuring the SSH Server, page 19
- Verifying the SSH Server, page 20

Configuring Password Protection

To session into the service module for direct SSH access to the Cisco AXP CLI and to configure the password, perform the following steps.

1. Session into the service module and configure the password.
2. **service password-encryption** password encrypts the clear text password that is entered and saves it in the encrypted format in the configuration file.

SUMMARY STEPS

1. **configure terminal**
2. **service password-encryption**

   or,

   no service password-encryption
3. **username sysadmin password** *clear-password-string*

   or,

   username sysadmin password 0 *clear-password-string*

   or,

   username sysadmin password 7 *hashed-password-string*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`service password-encryption`	Encrypts the clear text password using a system provided two-character salt string. The password is saved in the encrypted format in the configuration file. A clear text password string is encrypted and displayed as a level 7 password string.
	`no service password-encryption`	To keep the clear text password, then disable the **service password-encryption** command with the **no** prefix.
Step 3	`username sysadmin password` *clear-password-string*	You can use either one of these three commands: Specifies a non encrypted (cleartext) user password.
	`username sysadmin password 0` *clear-password-string*	Specifies a non encrypted password.
	`username sysadmin password 7` *hashed-password-string*	Specifies a hidden password.
Step 4	`exit`	Exits configuration mode.

Configuring the SSH Server

To configure SSH access to the CLI server allowing remote access to the Cisco AXP service module, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **ip ssh server**
3. **ip ssh interface** *interface*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`ip ssh server`	Enables the SSH service (starts the sshd daemon). Default SSH service is enabled. If the sshd daemon is running when you configure the **no** form of the command: the sshd daemon stops, the existing SSH session remains alive, and no new SSH sessions are accepted.
Step 3	`ip ssh interface` *`interface`*	Specifies the interface on which sshd should listen for an incoming connection. If you do not use this command, sshd listens on all interfaces. If sshd is configured to listen to a specific network interface, an IP address change for that interface prevents SSHD from accepting a connection on that modified interface. You must restart sshd or the Cisco AXP service module to re-establish SSH service.
Step 4	`exit`	Exits global configuration mode.

Verifying the SSH Server

To verify the SSH service is running, perform the following steps.

SUMMARY STEPS

1. **show processes**
2. **show processes memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`show processes`	View the state of the SSHD process. For this command and other diagnostic commands see Table 4.
Step 2	`show processes memory`	In EXEC mode, display the information of the sshd process when it is running.

Configuring the Syslog Server

Configuration of a syslog server allows the Cisco AXP service module to collect log messages from other physical and virtual devices on the network. The syslog server binds to an interface to accept log messages from any source on the network.

A user can enable or disable the syslog server on the Cisco AXP service module, and can specify the maximum log file size limits that can occupy the local file system space.

Because the syslog server cannot be configured to filter logs based on facility or priority, all log messages must be filtered before they are sent to the syslog server.

Log files generated by the syslog server reside in the */var/remote_log* directory, and the log file is named *remote_messages.log*. Rotated log files are appended with a number, such as *remote_messages.log.1*, with the higher numbers designating older files. The oldest log file is deleted during a file rotation.

SUMMARY STEPS

1. **configure terminal**
2. **syslog-server**
3. **syslog-server limit file-rotation** *size* [**file-size** *num*]
4. **syslog-server limit file-size** *size* [**file-rotation** *num*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters configuration mode.
Step 2	**`syslog-server`**	Enables or disables the syslog server. The syslog server is disabled by default. If the server is enabled, the Cisco AXP service module is used as a syslog server to receive all the log files from external devices. A error message: `ERROR - system does not have enough disk space` appears if: • The system has less than 80G disk storage, or, • Available disk space does not satisfy the current limits set by file size, and the number of files. Resolve error by either unloading applications to free disk space, or by changing limits. If this error occurs you cannot enable the syslog server.

	Command or Action	Purpose
Step 3	**syslog-server limit file-rotation** *size* [**file-size** *num*]	Sets syslog server limits. **file-rotation**-—Defines the number of log files to be rotated. The range is 1 to 40 and the default is 10. **file-size**—Defines the maximum size (in MB) of each log file. The range is 1 to 1000 MB and the default is 20 MB. Setting the file rotation configuration lower than the current settings causes extra log files to be deleted. **Example** If the current file rotation value is 5 and the new file rotation value is 2, log files 3 to 5 are deleted. A message, `WARNING - setting the new file-rotation value to 2 from the old value of 5 caused extra log files to be removed` notifies the user if they have specified a new file rotation value that is lower than the current file rotation value. Error messages: **Error Message** `System does not have enough disk space.` This error occurs if the available system disk space is not enough to satisfy the new configured limits. The file rotation and file size error messages appear if you enter an invalid value for a configuration. For example if you enter 80001 as the file-size or 99 as the file-rotation. The invalid values are rejected and the original limit values remain effective. **Error Message** `File-rotation is out of range (1-10).` **Error Message** `File-size is out of range (1-80000).`

	Command or Action	Purpose
Step 4	**`syslog-server limit file-size`** *`size`* `[`**`file-rotation`** *`num`*`]`	Sets syslog server limits. **file-rotation**-—Defines the number of log files to be rotated. The range is 1-40 and the default is 10. **file-size**— Defines the maximum size (in MB) of each log file. The range is 1-1000MB and the default is 20MB. Setting the file rotation configuration lower than the current settings causes extra log files to be deleted. **Example** If the current file rotation value is 5 and the new file rotation value is 2, log files 3 to 5 will be deleted. A message, `WARNING - setting the new file-rotation value to 2 from the old value of 5 caused extra log files to be removed` notifies the user if they have specified a new file rotation value that is lower than the current file rotation value. Error messages: **Error Message** `System does not have enough disk space.` This error arises if the available system disk space is not enough to satisfy the new configured limits. The file rotation and file size error messages are displayed if an invalid value is entered for a configuration. For example if you enter 80001 as the file-size or 99 as the file-rotation. The invalid values are rejected and the original limit values remain effective. **Error Message** `File-rotation is out of range (1-10).` **Error Message** `File-size is out of range (1-80000).`
Step 5	**`exit`**	Exits configuration mode.

Verifying the Syslog Server

To verify the Syslog server status, perform the following step.

SUMMARY STEPS

1. **show syslog-server**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`show syslog-server`	Verify the server status. See the "Syslog Server Logs" section on page 84 in Verifying and Troubleshooting.

Configuring the Application Service Environment

The core and add-on packages must be installed before proceeding with the configuration tasks in this section.

This section consists of the following tasks:

- Starting or Stopping the Application Service, page 26
- Configuring External Network Interfaces, page 26
- Configuring the NTP Server Source, page 31
- Configuring the Hostname, page 34
- Configuring the Application Domain, page 35
- Configuring Console Access, page 37
- Configuring IP Static Routes, page 38
- Source-based IP Routing, page 39
- Remote Serial Device Configuration, page 49
- Packet Analysis, page 54
- Logging, page 59
- Setting Resource Utilization Limits, page 62
- SSH Access to a Virtual Instance, page 63
- Configuring OSGi, page 66
- Synchronizing Files, page 66
- CLI Plug-in Invocation, page 70
- Cisco IOS Service API Configuration, page 71
- Cisco IOS Event Notification, page 76

Starting or Stopping the Application Service

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **no shutdown**
4. **show state**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`app-service` *`application-name`*	Enters application service mode.
Step 3	`no shutdown`	Starts or stops the specified application service. **shutdown** gracefully shuts down the application service and changes the application service state to offline after successfully completing configuration tasks.
Step 4	`show state`	(Optional) View the application service status.
Step 5	`end`	Exits configuration mode.

Configuring External Network Interfaces

Each application service environment is configured by default with access to the **eth0** network interface, and the loopback interface which automatically binds into the application environment.

Use the **bind interface** command to bind one or more active network interfaces to the application service environment. A network interface may be shared between application service environments.

Ensure that your configuration settings do not allow more than one application service environment to compete for the same port on the same networking device.

You can configure physical ethernet interfaces on the Cisco AXP service module through the CLI.

Derived network interfaces (such as subinterfaces and VLAN interfaces) must be activated before they are added to the list of available network interfaces.

Activation depends on the type of derived network interface. All physical and derived network interfaces have associated configuration commands to adjust IP and firewall settings.

Note The configuration of IP address cannot be changed for an RBCP controlled physical device—these settings are configured on the Cisco ISR.

Configuring Subinterfaces

Virtual and VLAN interfaces can only be created on a configured and nonvirtual interface. An appropriate route must be setup on the Cisco IOS software side to direct traffic to the new network.

In addition to configuring an appropriate route on the Cisco IOS software side to setup traffic to the VLAN interface, you must also configure the Cisco IOS software interface to DOT1Q mode.

DOT1Q mode only affects traffic that flows through this interface; it does not inject the VLAN tag for end-to-end traffic.

This section contains the following tasks:

- Attaching a Networking Device To/From the Application Environment, page 27
- Configuring a Subinterface for a Virtual Interface, page 29
- Configuring a Subinterface for a VLAN Interface, page 30

Attaching a Networking Device To/From the Application Environment

To attach a networking device to/from the application environment, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *device-name*
3. **ip address** *ip address network-mask*
4. **end**
5. **app-service** *application-name*
6. **bind interface** *network-interface-name*
7. **end**
8. **app-service** *application-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`interface`** *`device-name`*	Configures the network interfaces. *device-name*— Ethernet device name. For example, the device name can be **eth0 or eth1** for a built-in physical interface, **eth0:1** for a virtual interface, or **eth0.1** for a VLAN interface. • You can configure the virtual or VLAN interfaces only if these interfaces are not bound to the virtual hosting environment. If the interfaces are bound, an error message with the specific device name appears. For example, in the case of the interface **eth0.1** the following error message appears: **Error Message** `eth0.1 still bound to hosting environment(s), unbind first.` Do not remove a built-in physical interface. Upon removal, an error message appears: **Error Message** `Can not remove the built-in interface eth0/1.`
Step 3	**`ip address`** *ip-address* *`network-mask`*	Configures the IP address and network mask for the specified network interface. Changing the IP address for a bound interface results in a message warning the user that the application is bound to the interface. To remove the old IP configuration, reset the virtual instance.
Step 4	**`exit`**	Exits interface configuration mode.
Step 5	**`app-service`** *`application-name`*	Enters application service configuration mode.

	Command or Action	Purpose

Step 6	`bind interface` *`network-interface-name`*	Attaches or detaches a networking device to or from the application environment.
		network-interface-name— Interface name defined in the host, for example, the Ethernet *device-name* defined in the **interface** command.
		The interface is immediately available to the virtual instance with the execution of a new **bind** command.
		Removing an interface binding with the **no** prefix displays the following warning messages,
		`WARNING!!! Reset the hosting environment`
		`WARNING!!! For binding to be removed`
		and requires a virtual instance to restart.
		Note This command modifies configuration entries in the */etc/hosts* file for *ipaddr* and *hostname* mapping.
		ipaddr in the /etc/hosts file is modified when the command is issued. Only the first interface binding is used. Since **eth0** is the default to be bound for each virtual instance, *ipaddr* is normally eth0.

Step 7	`end`	Ends configuration mode.
Step 8	`app-service` *`application-name`*	Enters application service mode.
Step 9	`reset`	Resets the hosting environment.

Configuring a Subinterface for a Virtual Interface

To configure a subinterface for a virtual interface, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **interface eth0**.*x*
3. **ip address** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`interface eth0:`*x*	Configures the VLAN interface. Enters subinterface mode. *x*—Subinterface number.
Step 3	`ip address` *ip-address*	*Configures the IP address.*

Configuring a Subinterface for a VLAN Interface

To configure a subinterface for a VLAN interface, perform the following steps.

SUMMARY STEPS

1. On the router side:

   configure terminal

   ip routing

   interface integrated-service-engine *slot/port.x*

   encapsulation dot1q *vlanid*

   ip address *ip-address*

2. On the Cisco AXP Service Module:

   configure terminal

   interface eth *slot/port.x*

   ip address *ip-address*

DETAILED STEPS

	Command or Action	Purpose
	On the router side:	
Step 1	**configure terminal**	Enters configuration mode.
	ip routing	Enables IP routing on the router.
	interface integrated-service-engine *slot/port.x*	Enters subinterface mode. *x*—subinterface number
	encapsulation dot 1Q *vlanid*	Defines the encapsulation format as IEEE 802.1Q (dot1q), and specifies the VLAN identifier. vlanid—VLAN Identifier
	ip address *ip-address*	Configures the IP address for the interface.
	On the Cisco AXP service module:	
Step 2	**configure terminal**	Enters configuration mode.
	interface eth *slot/port.x*	Enters subinterface mode. *x*— Subinterface number
	ip address *ip-address*	Configures IP address.

Configuring the NTP Server Source

To have a consistent set of time stamps required for logs and development authorization between the router and the AXP service module, an administrator must configure the AXP service module to receive its NTP clock source from the router.

In the simplest case, the router is the master NTP server and uses its own system clock as the source for itself and the Cisco AXP service module. However, in many cases, the router will also use another trusted NTP server as its source.

SUMMARY STEPS

1. On the router:
   a. Configure the router as an NTP master server.

      Perform this step only if you do not have the router set to receive synchronization data from an external source.

      configure terminal

      ntp master

   b. Configure the correct time zone on the router.

      clock timezone *zone*

   c. Configure the router to broadcast the clock signal t to the integrated service engine.

      interface Integrated-Service-Engine 1/0

      ntp broadcast destination *ip address*

      end

      write memory

2. On the Cisco AXP service module:

   a. Configure the Cisco AXP service module's NTP server to point to the router as the source.

      configure terminal

      ntp server *ip address*

      end

      write memory

   b. Configure the time zone on the Cisco AXP service module to be the same as the router time zone. Reload the module for the new time zone to take effect.

      configure terminal

      clock timezone

      end

      write memory

      reload

DETAILED STEPS

	Command or Action	Purpose
	Configure the following commands on the router:	

Step 1	**`configure terminal`**
	`Example:`
	`Router# configure terminal`
		Enters global configuration mode.

	`ntp master`	Configures the router to use its own Network Time Protocol (NTP) master clock to synchronize with peers when an external NTP source is unavailable.
	`clock timezone`	*Sets the time zone on the router.*
	`interface Integrated-service-engine 1/0`	Enters interface configuration mode.

	`ntp broadcast destination` *`ip address`*
	`Example:`
	`Router(config-if) ntp broadcast destination 10.10.2.2`
		Creates an Network Time Protocol (NTP) broadcast server on a specified NTP interface.
		ip address—Destination host IP address. This is on the AXP module side of the integrated service engine.

	`end`	Exits global configuration mode.

	`write memory`
	`Example:`
	`Router# write memory`
		(EXEC mode) Writes the running configuration to the startup configuration.

	Configure the following commands on the AXP service module:	

Step 2	`configure terminal`
	`Example:`
	`SE-Module> configure terminal`	

	`ntp server` *`ip address`*	Configures the Network Time Protocol (NTP) server to keep the system time in synchronization with the NTP server. *ip address*—IP address of the NTP server providing the clock synchronization.
	`end`	Exits configuration mode.
	`write memory`	(Cisco AXP EXEC mode) Writes the running configuration to the startup configuration.
	`configure terminal`	Enters configuration mode.
	`clock timezone`	Sets the time zone on the AXP service module.
	`end`	Exits configuration mode.
	`write memory`	(Cisco AXP EXEC mode) Writes the running configuration to the startup configuration.
	`reload`	Allows the time zone to take effect.

Verifying NTP Server Configuration

Use the **show clock detail** command to verify the clock settings on the router and the AXP service module.

SUMMARY STEPS

1. On the router**:**

   show clock detail
2. On the AXP service module:

   show clock detail

DETAILED STEPS

	Command or Action	Purpose
	On the router:	

Step 3	**`show clock detail`**
	`Example:`
	`Router# show clock detail`
	`00:24:02.669 UTC Sat Oct 27 2007`
	`Time source is NTP`
		Displays clock settings on the router.

	`On the AXP service module:`	

Step 4	**`show clock detail`**
	`Example:`
	`se-Module> show clock detail`
	`16:43:30.616 PDT Fri Oct 26 2007`
	`time zone:`
	`America/Los_Angeles`
	`clock state: unsync`
	`delta from reference (microsec): 0`
	`estimated error (microsec): 16`
	`time resolution (microsec): 1`
	`clock interrupt period (microsec): 10000`
	`time of day (sec): 1193442210`
	`time of day (microsec): 619436`
		Displays clock settings on the AXP service module.

Configuring the Hostname

Configure the hostname for the application using the commands shown below.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **hostname** *hostname*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**app-service** *application-name*	Enters application service mode.

Step 3	**hostname** *hostname*	Configures the hostname for the application.
		The hostname is limited to 32 characters.
		If you enter more than 32 characters an error message appears:
		Error Message `hostname size greater than 32`
		If the hostname is not configured, the default hostname on the host side is used.
		This command modifies configuration directives in /etc/hosts and updates the hostname of the hostname-ip mapping entry. It creates the */etc/hosts* file and adds the next entry in it, if the file does not exist.
		If an application package has already bundled its own /etc/hosts file, the new entries are appended to the existing entries and the original entries remain intact.
		The ipaddr in the */etc/hosts* file is modified when you use the **bind interface** command. The first interface binding is used.
		The ipaddr is usually eth0 because eth0 is the default, and is bound to each virtual instance.
		Example:
		`/etc/hosts:`
		`10.0.0.1 localhost.localdomain  localhost     ## added by cli`
		`ipaddr hostname.domain     hostname    ## added by cli`

Configuring the Application Domain

Using the **ip name-server** and **ip domain-name** commands in the host populates the */etc/resolv.conf* file in each installed virtual instance. Using these commands to change the configuration in the host results in the */etc/resolv.conf* file being updated.

When these commands are used to configure a new name-server and domain-name for a virtual instance (in app-service mode), the */etc/resolv.conf* file in that virtual instance is overridden with the new server name and domain name.

The */etc/resolv.conf* file in that virtual instance reverts back to the host configuration whenever the virtual instance does not have a name-server or domain-name configured.

Configuring the name-server and domain-server in a virtual instance always takes precedence over configuration in the host.

DNS Address

To configure the address of the domain name server (DNS) for the application, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **ip name-server** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`app-service`** *`application-name`*	Enters application service mode.

Step 3	**`ip name-server`** *`ip-address`*	• *ip-address*— IP address of the DNS.
		• Configures the IP address of the domain name server (DNS) for the application.
		• A maximum of two DNS servers can be defined.
		In a Linux environment, the */etc/resolv.conf* file typically contains the IP addresses of name servers (DNS name resolvers) that attempt to translate names into addresses for any node available on the network.
		This command creates the */etc/resolv.conf* file and adds the name-server entries in it if the file does not exist.
		If an application package has already bundled its own */etc/resolv.conf* file, the new entries are appended to the existing ones and will leave the original ones intact.
		Example:
		`search localdomain## added by cli`
		`domain localdomain## added by cli`
		`nameserver x.x.x.x## added by cli`

Domain Name

To configure the domain name for the application, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **ip domain-name** *domain name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`app-service`** *`application-name`*	Enters application service mode.

Step 3	**`ip domain-name`** *`domain name`*	Configures the domain name for the application.
		The domain-name is limited to 64 characters.
		If more than sixty four characters are entered, the following error message is displayed:
		Error Message `domain size greater than 64`
		Default: The domain name is not configured.
		This command modifies configuration directives in */etc/hosts* and */etc/resolv.conf* files where the domain name is relevant.
		This command also modifies the search list for hostname lookup and domain directives for local domain name in */etc/resolv.conf* file.
		For */etc/hosts* file, it updates the domain name of the hostname-ip mapping entry.
		Example:
		`/etc/resolv.conf:`
		`search cisco.com    ## added by cli`
		`domain cisco.com    ## added by cli`
		`nameserver x.x.x.x  ## added by cli`
		`/etc/hosts:`
		`10.100.50.10 appre.cisco.com appre`

Configuring Console Access

To configure console access, perform the following steps.

SUMMARY STEPS

1. **app-service** *application-name*
2. **connect console**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`app-service`** *`application-name`*	Enters application service mode.
Step 2	**`connect console`**	Allows a third party to integrate their own application commands to the console shell. On initiating the command, **/bin/console** executes. The third party application must provide its own console file in binary or as a script (telnet to their CLI), to cross connect to their CLI shell. If the application does not provide a console file, the following message appears: **Error Message** `Unable to start console`

Configuring IP Static Routes

Cisco AXP uses statically configured routes to route traffic to a specific interface. The destination network prefix, network prefix mask and the gateway address are in IPV4 dotted notation (xx.xx.xx.xx).

You can specify multiple route lines in a configuration and use the **show ip route** command to display the configured routes saved in the database. Some routes are added by default when configuring the interface. The default route for **eth0** is configured through the Cisco IOS software CLI.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *device-name*
3. **ip route** *destination-network-prefix network-prefix-mask gateway*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`interface`** *`device-name`*	Configures the network interfaces
Step 3	**`ip route`** *`destination-network-prefix network-prefix-mask gateway-address`*	Configures the IP route. *destination-network-prefix*— IP route prefix for the destination *network-prefix-mask*— Prefix mask for the destination *gateway-address—IP address of the gateway.*

Source-based IP Routing

Source-based IP routing, also known as static route configuration, is necessary for application initiated data transfer, such as client applications, and is used to determine an outbound interface when multiple interfaces are bound to an application instance.

Source-based routing is implemented for server applications to route response packets back through the incoming interface, and it is independent of the destination address.

Consider traffic entering the Cisco AXP service module through an ethernet interface, for example eth0.20, from an external IP address X. When the Cisco AXP application generates a reply, the system now contains a packet with source IP address, which is the address for eth0.20, and the destination IP address X.

If source-based routing is not applied, this packet is sent to a default route through eth0. Source based routing routes traffic based on the source IP address and sends it through the originating interface, which in our example above, is eth0.20.

Note For the Cisco AXP network configuration, the destination interface to which you send the response packet is the same as the incoming interface.

If an application specifies the source IP address when a socket is opened, it will use source-based routing to select the interface to send traffic.

Access Control List

Configuring an access control list (ACL) on the Cisco AXP platform is similar to configuring an ACL on Cisco IOS software.

Packet filtering helps control packet movement through the network by helping to limit network traffic and restrict network use by certain users or devices. Use ACLs to permit or deny packets from crossing specified interfaces.

Using the **ip access-list standard** command enables standard ACL configuration mode (config-std-nacl). You can then configure the **permit** command in ACL sub-mode (config-std-nacl) to set up the standard IP access list.

Note In Cisco AXP 1.0.1, you can specify only one IP address in the access control list.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list standard** *{acl-name* | *acl-num}*
3. [*line-num*] **permit** {*source-ip* [*wildcard*]| **host** *source-ip* | **any**}[**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** SE-Module> configure terminal	Enters global configuration mode.
Step 2	**ip access-list standard** *{acl-name | acl-num}* Example: se_module (config)> **ip access-list standard test**	Enables standard ACL configuration mode (config-std-nacl). This command enters standard ACL configuration mode in which all subsequent commands apply to the current standard access list. *acl-name*—Access list to which all commands entered from ACL configuration mode apply, using an alphanumeric string of up to 30 characters, beginning with a letter. *acl-num*— Access list to which all commands entered from access list configuration mode apply, using a numeric identifier. For standard access lists, the valid range is 1 to 99.
Step 3	[*line-num*] **permit** {*source-ip* [*wildcard*]| **host** *source-ip*|**any**}[**log**] se-Module (confg-std-nacl)> **permit 155.168.10.0 any**	Adds a line to a standard access-list that specifies the type of packets to be permitted for further processing. The **permit** command is used in standard ACL configuration mode (config-std-nacl). *line-num*— Entry at a specific line number in the access list. **permit**—Allows packets that match the specified conditions to be processed. *source-ip*—Source IP address. Number of the network or host from which the packet is being sent, specified as a 32-bit quantity in 4-part dotted-decimal format (for example, 0.0.0.0). *wildcard*—(Optional) Portions of the preceding IP address to match, expressed using 4-digit, dotted-decimal notation. Bits to match are identified by a digital value of 0; bits to ignore are identified by a 1. For standard IP ACLs, the wildcard parameter of the **ip access-list** command is always optional. If the host keyword is specified for a standard IP ACL, then the wildcard parameter is not allowed. **host**— Matches the next IP address. **any**—Matches any IP address. **log**—(Optional) Sends a logging message to the console about the packet matching the entry. The message includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets. The message is generated for the first packet that matches the entry, and then repeats at 5-minute intervals, including the number of packets permitted or denied in the previous 5-minute interval.

Verifying Access Control Lists

To use the **show ip access-list** command in Cisco AXP EXEC mode to view the access control lists configured on the platform, perform the following step.

SUMMARY STEPS

1. **show ip access-list** [<**1-99**> | <*name*>][**interface** *intf*] [**details**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip access-list** [<**1-99**>|<*name*>][**interface** <*intf*>][**details**] `SE-Module>show ip access-list`	List the rule set of an access-list specified by number or name. It also lists the access-list associated with a specific interface. If a name or number of the interface is not entered, the command lists the entire rule-set of all the access lists configured in the system. **1-99**—Access list number *name*—Access list name *intf*—Interface name. **details**—The raw iptable format of display will be used to display the chain created by the ACL list.

Configuring Source Based Routing

Prerequisites for Configuring Source Based Routing

- Configure the routing/forwarding (VRF) tables. See Configuring VRF-Lite.

SUMMARY STEPS

1. Configure the following Cisco IOS software commands on the router:

   configure terminal

   ip vrf *vrf-name*

   rd *ip-address*

   route-target export *ip-address*

   route-target import *ip-address*

   interface GigabitEthernet0/1

   ip address *ip-address network-mask*

   duplex auto

   speed auto

   ip vrf forwarding *vrf-name*

```
interface Integrated-Service-Engine1/0
 ip unnumbered GigabitEthernet0/0
 service-module ip address ip-address network-mask
 service-module ip default-gateway ip-address
 no keepalive

interface integrated-service-engine 1/0.1
encapsulation dot 1q vlan-id
ip address ip-address network-mask
ip vrf forwarding vrf-name
exit
```

2. Configure the following AXP commands on the service module:

```
configure terminal
interface device-name
 ip address ip-address network-mask
 ip route table table-num
exit
ip access-list standard {acl-name | acl-num}
[line-num] permit {source-ip [wildcard]| host source-ip|any}[log]
route-map name number
 match ip address {acl-num | acl-name }
 set route table table-num
exit
ip local policy route-map map-tag
ip route table num dest-prefix net-mask default-gw
ip route table num dest-prefix net-mask blackhole
exit
```

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure the following Cisco IOS software commands on the router:	

	`configure terminal`
	`Router# configure terminal`
		Enters global configuration mode.

	`ip vrf` *`vrf-name`*	Configures a VRF routing table and enters VRF configuration mode. *vrf-name*— Name assigned to a VRF.

	`rd` *`route-distinguisher`*
	`Router(config-vrf)# rd 200.7.7.1:10`
		Adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.

	`route-target export` *`ip-address`*	Exports routing information from the target VPN extended community.
	`route-target import` *`ip-address`*	Imports routing information from the target VPN extended community.
	`exit`	Exits VRF configuration mode.
	`interface GigabitEthernet 0/1`	Selects an interface to configure and enters interface configuration mode.
	`ip address` *`ip-address network-mask`*	Selects the IP address.
	`duplex auto`	Configures the duplex operation on an interface. **auto**—Specifies the autonegotiation capability. The interface automatically operates at half or full duplex, depending on: – Environmental factors, such as the type of media. – Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
	`speed auto`	Configures the speed for a Fast Ethernet interface. **auto**—Turns on the Fast Ethernet autonegotiation capability. The interface automatically operates at 10 or 100 Mbps depending on: – Environmental factors, such as the type of media. – Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
	`ip vrf forwarding` *`vrf-name`*	Associates a VRF with an interface or subinterface. *vrf-name*—Name assigned to a VRF.
	`interface Integrated-Service-Engine 1/0`	Selects an interface to configure and enters interface configuration mode.
	`ip unnumbered GigabitEthernet0/0`	The ip unnumbered command enables IP processing on an interface without assigning an explicit IP address to the interface.
	`service-module ip address` *`ip-address network-mask`*	Specifies the IP address for the module interface to the router.

	Command or Action	Purpose
	service-module ip default-gateway *ip-address*	Specifies the IP address for the default gateway router for the module.
	no keepalive	Disables the ability to send keepalive packets.
	interface integrated-service-engine 1/0.1 Example: Router(config)# interface integrated-service-engine 1/0.1	Enters sub-interface mode.
	encapsulation dot 1q *vlan-id* Example: Router(config-subif)# encapsulation dot 1q 10	Configures the subinterface as a VLAN subinterface. **dot1q**—defines the encapsulation format as IEEE 802.1Q VLAN. *vlanid*—number that identifies the VLAN. The router applies the service policy of the physical interface to all of the individual VLANs configured on the interface.
	ip address *ip-address network-mask* Example: Router(config-subif)# ip address 10.7.8.8 255.255.255.0	Sets the IP address of the interface.
	ip vrf forwarding *vrf-name* Example: Router(config-subif)# ip vrf forwarding red	Configures the VRF forwarding table. *vrf-name*—VRF table name.
	end	Exits configuration mode.
Step 2	Configure the following Cisco AXP commands on the service module:	
	configure terminal	Enters global configuration mode.
	interface *device-name*	Configures the network interfaces. *device-name*— Ethernet device name For example, the device name can be **eth0 or eth1** for a built-in physical interface, **eth0:1** for a virtual interface, or **eth0.1** for a VLAN interface. • You can configure the virtual or VLAN interfaces only if these interfaces are not bound to the virtual hosting environment.
	ip address *ip-address network-mask*	Sets the IP address.
	ip route table *table-num*	Sets up the connected route. *table-num*— Select a route table number from 1 to 100.
	exit	Exits interface mode.

Command or Action	Purpose
ip access-list standard {*acl-name* | *acl-num*}	Enables standard ACL configuration mode (config-std-nacl). This command enters standard ACL configuration mode in which all subsequent commands apply to the current standard access list. *acl-name*—Access list to which all commands entered from ACL configuration mode apply. using an alphanumeric string of up to 30 characters, beginning with a letter. *acl-num*— Access list to which all commands entered from access list configuration mode apply, using a numeric identifier. For standard access lists, the valid range is 1 to 99. You can set further options under standard ACL configuration mode (config-std-nacl) as shown in the remaining steps.

Command or Action	Purpose
`[line-num] permit {source-ip [wildcard]| host source-ip|any}[log]`	Configured in access-list configuration mode. Adds a line to a standard access-list that specifies the type of packets to be permitted for further processing. Use the **permit** command in standard ACL configuration mode (config-std-nacl). *line-num (optional)*— Entry at a specific line number in the access list. **permit**—Allows packets that match the specified conditions to be processed. *source-ip*—Source IP address. The number of the network or host from which the packet is being sent, specified as a 32-bit quantity in 4-part dotted-decimal format (for example, 0.0.0.0). *wildcard*—(Optional) Portions of the preceding IP address to match, expressed using 4-digit, dotted-decimal notation. Bits to match are identified by a digital value of 0; bits to ignore are identified by a 1. **Note** For standard IP ACLs, the wildcard parameter of the **ip access-list** command is always optional. If the host keyword is specified for a standard IP ACL, then the wildcard parameter is not allowed. **host**— Matches the following IP address. **any**—Matches any IP address. **log**—(Optional) Sends a logging message to the console about the packet matching the entry. The message includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets. The message is generated for the first packet that matches the entry, and then repeats at 5-minute intervals, including the number of packets permitted or denied in the previous 5-minute interval.
`route-map map-tag number`	Enters route map configuration mode. The route map is used to match source filtering with a specific routing table. *map-tag*—Select a name for the route map. *number*—Select a route map number from 1 to 100.
`match ip address {acl-num|acl-name }`	Matches the IP address for the route map using either the number or the name of the access control list. *acl-num*—Access control list number. *acl-name*—Name of the access control list.

Command or Action	Purpose
set route table *table-num*	Sets the route table. *table-num*— Same table number as in the **ip route table** command.
exit	Exits route-map subcommand mode.
ip local policy route-map *map-tag*	Identifies a route map to use for policy routing. *map-tag*—Name must match the map-tag in the **route-map** command.
ip route table table-*num* *dest-prefix* *net-mask* *default-gw*	Sets the route table for a specific destination prefix and default gateway. *table-num*—Same table number as in the **ip route table** command. *dest-prefix*—Destination prefix *net-mask*—Network mask *default-gw*—Default gateway
ip route table table-*num* *dest-prefix* *net-mask* **blackhole**	*able-num*—Same table number as in the **ip route table** command. *dest-prefix*—Destination prefix *net-mask*—Network mask *default-gw*—Default gateway **blackhole**—Sets a blackhole route for dropping packets.
exit	Exits global configuration mode.

Examples

Source Based IP Routing

```
interface eth0.100
  ip route table 10   <-- sets up the connected route for table 10
  ip address 10.7.8.9 255.255.255.0
  exit
Interface eth0.200
  ip route table 20
  ip address 11.11.10.2 255.255.255.0
  exit
ip access-list standard 100
  permit 10.7.8.9        <-- Source address that will be used for source based routing
  exit
ip access-list standard 200
  permit 11.11.10.2
  exit
ip route table 10 0.0.0.0 0.0.0.0 10.7.8.10   <--- defines the default route in table 10
ip route table 20 0.0.0.0 0.0.0.0 11.11.10.3
route-map CLASSIFY 10
  match ip addr 100   <--- defines source based routing address and routing table.
  set route table 10
  exit
route-map CLASSIFY 20
  match ip addr 200
  set route table 20
  exit
ip local policy route-map CLASSIFY
```

VRF Configuration

In this example, the VRF is named red, and dot1Q encapsulation is used with ID tag 10 to relay VRF traffic from the router to the service module.

```
ip vrf red
 rd 192.0.2.0:10
 route-target export 192.0.2.0:10
 route-target import 192.0.2.0:10
interface GigabitEthernet0/1
 ip address 10.7.7.7 255.255.255.0
 duplex auto
 speed auto
 ip vrf forwarding red
interface Integrated-Service-Engine1/0
 ip unnumbered GigabitEthernet0/0
 service-module ip address 209.165.201.1 255.255.255.224
 service-module ip default-gateway 209.165.201.2
 no keepalive
interface Integrated-Service-Engine1/0.1
 encapsulation dot1Q 10
 ip address 10.7.8.8 255.255.255.0
 ip vrf forwarding red
```

Remote Serial Device Configuration

Cisco AXP supports external device connections through Cisco IOS software host serial ports. The virtual serial device on the local Cisco AXP platform interacts with external Cisco IOS software serial devices.

When the virtual serial device opens, a reverse telnet session is established, connecting to the Cisco IOS software host line interfaces. All serial data transfer is carried through this reverse telnet session.

Linux applications use the virtual device driver to control and receive signal state notifications on all async RS232 leads. A fixed TCP port is assigned to each of the TTY and AUX lines, and to the serial interfaces in ASYNC mode, when reverse telnet is implemented in Cisco IOS software.

The port-index and the TCP port for various interfaces are pre-defined in Cisco IOS software, and are shown in Table 2.

Table 2 Cisco IOS Software Line Numbers and Port Values

Interface Name	Port Index	TCP Port
Console	0	Cannot be used.
tty1	1	6001
tty2	2	6002
...	...	...
ttyn	n	600n
AUX	n+1	600(n+1)
vtty1	n+2	600(n+2)

Table 2 Cisco IOS Software Line Numbers and Port Values

Interface Name	Port Index	TCP Port
...	...	...
Serial interface in ASYNC mode	Platform dependent	Platform dependent

PREREQUISITE TASKS

- The vserial add-on package must be first installed, followed by an installation of the third party application before commencing with the following configuration.
- See the *Cisco AXP Developer Guide* for information on application packages.
- To configure the virtual serial device driver, you must configure:
  - *NETCONF over BEEP*
  - Reverse telnet on the router and the Cisco AXP service module.

Guidelines from RFC 2217 are also used in implementing the virtual serial device driver on the Cisco AXP platform:

Telnet Com Port Control Option document RFC 2217)

SUMMARY STEPS

1. Configure the following on the router:

   configure terminal

   sasl profile *profile-name*

   mechanism *profile-mechanism*

   exit

   netconf max-sessions *session-number*

   netconf beep listener [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*]

2. Configure the serial device interface.

   interface serial *slot/module/port*

   physical-layer async

   no ip address

   encapsulation slip

3. Configure lines for reverse telnet.

   line con *line-num*

   exec-timeout 0 0

   login local

   stopbits 1

   line 0/0/0

   transport input telnet

   line *line-num*

```
no activation-character
no exec
transport preferred none
transport input all
transport output pad telnet rlogin lapb-ta mop udptn v120
```

4. Configure the following on the Cisco AXP service module.

   ```
   username user-name password password
   netconf max-sessions session-number
   netconf beep initiator router-IP-address port-number
   ```

5. Bind the interface and serial devices.

   ```
   conf t
   app-service serialapp
   bind interface
   bind serial
   end
   app-service serialapp
   reset
   end
   ```

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure on the router side:	
	configure terminal Example: router# configure terminal	Enters configuration mode.
	sasl profile *profile-name*	Sets the SASL profile.
	Router(config-sasl-profile)# **mechanism anonymous**	Configures the SASL profile mechanism
	exit	Exit SASL profile mode.
	netconf max-sessions *num*	Specifies the maximum number of concurrent NETCONF sessions. *num*—maximum number of sessions.
	netconf beep listener [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*]	Specifies BEEP as the transport protocol for NETCONF and configures a peer as the BEEP listener.
Step 2	**Configure the serial device interface:**	
	configure terminal	Enters configuration mode
	interface serial *slot/subslot/port*	Defines the interface serial parameters.
	physical-layer async	Configures the interface for async communication.
	no ip address	No ip address is set.
	encapsulation slip	Configures slip encapsulation.
Step 3	*Configure lines for reverse telnet:*	
	line con *line-num*	Configures the console terminal line. *line-num*— Line number
	exec-timeout *minutes* [*seconds*]	Sets the interval for the EXEC command interpreter to wait, until user input is detected.
	login local	Enables local username authentication
	stopbits 1	Sets the number of stop bits transmitted per byte. **1**— One stop bit.
	line *Slot/Subslot/Port*	Defines line parameters.
	transport input telnet	Defines the protocols for connection to a specific line of the router. **telnet**—Specifies all types of incoming TCP/IP connections
	line *num*	Selects the specific line. *num*—Line number
	no activation-character	Makes any character activate a terminal.
	no exec	Turns off the EXEC process for the specified line in the l**ine** command.

	Command or Action	Purpose
	transport preferred none	Specifies the transport protocol that the Cisco IOS software uses if the user does not specify one when initiating a connection. **none**— Prevents any protocol selection on the line. The system normally assumes that any unrecognized command is a hostname. If the protocol is set to **none**, the system no longer makes that assumption. No connection is attempted if the command is not recognized.
	transport input all	Defines the protocols for connection to a specific line of the router. **all**—Selects all protocols.
	transport output pad telnet rlogin lapb-ta mop udptn v120	**pad—**Selects X.3 PAD, used most often to connect routers to X.25 hosts. **telnet** —Selects the TCP/IP Telnet protocol. It allows a user at one site to establish a TCP connection to a login server at another site. **rlogin**—Selects the UNIX rlogin protocol for TCP connections. The rlogin setting is a special case of a Telnet connection. If an rlogin attempt to a particular host has failed, the failure will be tracked, and subsequent connection attempts will use a Telnet connection. **lapb-ta**— Selects Link Access Procedure, Balanced-Terminal Adapter (LAPB-TA) **mop**— Selects Maintenance Operation Protocol (MOP). **udptn**— Selects User Datagram Protocol (UDP) Telnet (UDPTN) connections. **v120**— Select the V.120 protocol for outgoing asynchronous over ISDN connections.
Step 4	Configure the following on the Cisco AXP Service Module:	
	server *user-name* **password** *password*	Configures a SASL server.
	netconf max-sessions *session-number*	Specifies the maximum number of concurrent NETCONF sessions allowed.
	netconf beep initiator *router-IP-address port-number*	Specifies BEEP as the transport protocol for NETCONF sessions and configures a peer as the BEEP initiator.
Step 5	Bind the interface device, and bind the serial device to *serialapp,* and reset.	
	configure terminal	Enters configuration mode.
	app-service *serialapp*	Enters application service mode. *serialapp*—Serial device application name inside the third party application.
	bind interface *network-interface-name*	Attaches the networking device to or from the virtual environment.

Command or Action	Purpose
bind serial *device-id* [*device-id-on-hosting environment*] **Example:** **bind serial vtty000 modem**	Binds the serial device, which is connected to the Cisco IOS software side, inside the virtual environment. *device-id*— Device ID of the serial device connected to the Cisco IOS software side. Use the **show device serial** command to view device ID. *device-id-on-hosting-environment*—(Optional) Designates a name that is different from the device ID (*device-id*) inside the hosting environment.
end	Exits application service mode.
app-service *serialapp*	Enters application service mode. *serialapp*— Serial device application name inside the third party application.
reset	Resets the application service environment.
end	Exits application service mode.

Packet Analysis

You can use either the network analysis module (NAM), or the router IP traffic (RITE) feature for packet monitoring on the Cisco AXP platform. Both features are very similar to the features available on Cisco IOS software platforms.

Depending on your application, and the traffic analysis required for that application, these suggestions to help you choose the appropriate feature for your application:

- If your application requires monitoring router-generated packets, use NAM because RITE does not support this type of monitoring.
- For a quick start, and if your application does require monitoring of router generated packets, use NAM.
- The RITE feature is best if:
  - You need more granular control on your exported traffic.
  - Router generated packets are not a requirement for your application.

Note Both features can be configured at the same time, however, we do not recommend simultaneous feature configuration. Each feature performs its own duplication, resulting in receipt of duplicated packets if both features are configured at the same time.

For more information on these features, see Cisco Branch Routers Series Network Analysis Module and RITE documents on cisco.com. Also see the “Additional References” section on page 99.

Configuring Router IP Traffic Export

The advantages of using RITE include:

- Lightweight export
- Can capture incoming or bidirectional traffic

- Allows user to configure access control lists (ACLs)
- Allows user to configure sampling rate

The following RITE configurations are on the router side:

SUMMARY STEPS

1. **configure terminal**
2. **ip traffic-export profile** *profile-name*
3. **bidirectional**
4. **interface Integrated-Service-Engine** *slot/***0**
5. **mac-address** *H.H.H*
6. **incoming** {**access-list** {standard | extended | named} | **sample one-in-every** *packet-number*}
7. **outgoing** {**access-list** {standard | extended | named} | **sample one-in-every** *packet-numbe*r}
8. **exit**
9. **interface** *type number*
10. **description** *string*
11. **ip traffic-export apply** *profile-name*
12. **duplex auto**
13. **speed auto**
14. **no keepalive**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** Example: Router# configure terminal	Enters global configuration mode.
Step 2	**ip traffic-export profile** *profile-name* Example: Router(config)# ip traffic-export profile corp1	Creates or edits an IP traffic export profile, enables the profile on an ingress interface, and enters RITE configuration mode.
Step 3	**bidirectional**	Exports incoming and outgoing IP traffic on the monitored interface. **Note** If this command is not enabled, only incoming traffic is exported.
Step 4	**interface Integrated-Service-Engine** *slot* **/0**	Enters interface configuration mode for the slot and port where the service module resides. • *slot*—Specifies the service module slot.
Step 5	**mac-address** *H.H.H*	Specifies the 48-bit address of the destination host that is receiving the exported traffic. Since the service module is set to route packets by default, configuring a valid MAC address causes the routing stack of the service module to route the traffic, resulting in duplicated packets being routed. To avoid this scenario, consider configuring an invalid MAC address which is different from the MAC address of the service module. Since the service module configuration is in promiscuous mode, an invalid MAC address will also work, allowing traffic to reach the service module without being routed.
Step 6	**incoming** {**access-list** {standard|extended| named}|**sample one-in-every** *packet-number*}	(Optional) Configures filtering for incoming traffic. **Note** After you create a profile with the **ip traffic-export apply** command, this functionality is enabled by default.
Step 7	**outgoing** {**access-list** {standard|extended| named}|**sample one-in-every** *packet-number*}	Configures filtering for outgoing export traffic. • If you use this command, you must also use the **bidirectional** command, which enables outgoing traffic to be exported. • Only routed traffic (such as pass through traffic) is exported. Traffic that originates from the network device is not exported.
Step 8	**exit**	(Optional) Exits RITE configuration mode.
Step 9	**interface** *type number* Example: Router(config)# intrerface GigabitEthernet 0/0	Selects an interface to configure and enters interface configuration mode. *type*—Type of interface to be configured. For example GigabitEthernet. *number*—Module and port number.

	Command or Action	Purpose
Step 10	**`description`** *`string`*	Adds a description to an interface configuration. *string*—Clue to help you remember what is attached to this interface. This string is limited to 238 characters.

Step 11	**`ip traffic-export apply`** *`profile-name`*
	`Example:`
	`Router(config-if)# ip traffic-export apply corp1`
		Creates or edits an IP traffic export profile, enables the profile on an ingress interface, and enters RITE configuration mode.

Step 12	**`duplex auto`**	(Optional) Configures the duplex operation on an interface. **auto**—Specifies the autonegotiation capability. The interface automatically operates at half or full duplex, depending on: • Environmental factors, such as the type of media. • Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
Step 13	**`speed auto`**	(Optional) Configures the speed for a Fast Ethernet interface. **auto**—Turns on the Fast Ethernet autonegotiation capability. The interface automatically operates at 10 or 100 Mbps depending on: • Environmental factors, such as the type of media. • Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
Step 14	**`no keepalive`**	Disables the ability to send keepalive packets.
Step 15	**`end`**	Exits interface mode.

Network Analysis Monitoring

Packet monitoring on the Cisco AXP platform is very similar to packet monitoring in a Cisco IOS software environment.

To enable packet monitoring on a Cisco AXP service module interface, use the **analysis-module monitoring** command in interface configuration mode.

The following NAM configurations are on the router side:

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **description** *string*
4. **ip address** *ip-address network-mask*
5. **duplex auto**
6. **speed auto**
7. **analysis-module monitoring**

8. **no keepalive**
9. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure terminal`**
	`Example:`
	`router# configure terminal`
		Enters global configuration mode.

Step 2	**`interface`** *`type number`*
	`Example:` **`interface GigabitEthernet0/0`**
		Selects an interface to configure and enters interface configuration mode.
		type— Type of interface to be configured. For example GigabitEthernet.
		number— Module and port number.

Step 3	**`description`** *`string`*
	`Example:`**`description $ETH-LAN$$ETH-SW-LAUNCH$$INTF-INFO-GE 0/0`**
		Adds a description to an interface configuration.
		string—Clue to help you remember what is attached to this interface. This string is limited to 238 characters.

Step 4	**`ip address`** *`ip-address network-mask`*	Configures the IP address and network mask for the specified network interface.
Step 5	**`duplex auto`**	(Optional) Configures the duplex operation on an interface. **auto**—Specifies the autonegotiation capability. The interface automatically operates at half or full duplex, depending on: • Environmental factors, such as the type of media. • Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
Step 6	**`speed auto`**	(Optional) Configures the speed for a Fast Ethernet interface. **auto**—Turns on the Fast Ethernet autonegotiation capability. The interface automatically operates at 10 or 100 Mbps depending on: • Environmental factors, such as the type of media. • Transmission speeds for the peer routers, hubs, and switches used in the network configuration.
Step 7	**`analysis-module monitoring`**	Enables packet monitoring on an interface.
Step 8	**`no keepalive`**	Disables the ability to send keepalive packets.
Step 9	**`end`**	Exits interface mode.

References

- *Cisco Branch Routers Series Network Analysis Module*
- *Cisco Network Analysis Module Software*

Logging

This section consists of:

- Configuring Log File Size Limits, page 59
- Configuring Remote Logging, page 60
- Configuring System Log Levels, page 61

Configuring Log File Size Limits

To configure the log file size, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **limit log-file size** *megabytes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`app-service` *application-name*	Enters application service mode.
Step 3	`limit log-file size` *megabytes*	- Sets the size of the log file /var/log/*messages.log*. Each virtual instance writes a syslog to its own file */var/log/messages.log*. - Once this file reaches the limit specified by this command, its contents are moved to a backup log file *messages.log.prev* and a new *messages.log* file is started. The range is 0 to 40MB with a default size of 5MB for two files. - When the log file size reaches its limit, messages are moved to an alternate file *messages.log.prev.* - *megabytes*—The range of the log file size from 0 - 40 MB. - When the value is out of range, the following message appears: `%Invalid input detected at '^' marker` - If the log file size limits are not set (**no limit log-file size**), the size reverts to the default value of 5MB. - If the log file size is set to 0 MB, a minimum file size of 10 KB is set. To view log files under the */var/log* directory, use the **show log name** command in the "Viewing Log and Core Files" section on page 81.

Configuring Remote Logging

To configure remote logging, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *device-name*
3. **no shutdown**
4. **ip address** *ip address network-mask*
5. **end**
6. **app-service** *application-name*
7. **log server address** *hostname*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`interface device-name`**	Configures the network interfaces.
Step 3	**`no shutdown`**	Starts the specified application service.
Step 4	**`ip address`** *`ip-address network-mask`*	Configures the IP address and network mask for the specified network interface.
Step 5	**`end`**	Exits interface configuration mode.
Step 6	**`app-service`** *`application-name`*	Enters application service mode.
Step 7	**`log server address`** *`hostname`*	Enables remote logging and configures the remote logging server. Application syslog messages are sent to the specified log server. The hostname can be an IP address or hostname. When you enter an invalid IP address such as 0.0.0.0, the following error message appears: **Error Message** `0.0.0.0 is an invalid Host IP address`

Configuring System Log Levels

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **log level** *levels*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`app-service`** *`application-name`*	Enters application service mode.

Step 3	**`log level`** *`levels`*
	`Example:`
	`SE-Module(config-app-service)>` **`log level info`**
		Configures the system log level.
		Applicable **levels** are:
		info— Events with LOG_INFO and higher severity are logged, including all messages described in **notice**.
		warn (Default)—Events with LOG_WARNING and higher severity are logged, including all error messages described in **err**.
		err—Events with LOG_ERR and higher severity are logged, including LOG_EMERG, LOG_ALERT, and LOG_CRIT.
		notice —Events with LOG_NOTICE and higher severity are logged, including all messages described in **warn**.
		debug—Events with LOG_DEBUG and higher severity are logged, including all messages described in **info**.

Setting Resource Utilization Limits

To set disk utilization and CPU utilization limits, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. **limit disk utilization** *Megabytes*
4. **limit cpu utilization** *index*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure terminal`**	Enters global configuration mode.
Step 2	**`app-service`** *`application-name`*	Enters application service mode.
Step 3	**`limit disk utilization`** *`Megabytes`*	Modifies the disk utilization setup during installation. It will take effect the next time the application instance has restarted. *Megabytes*: 1–100000 MB The disk utilization range varies between the minimum limit specified by the package to the maximum limit available by the system.
Step 4	**`limit cpu utilization`** *`index`*	Modifies the CPU utilization limit during application installation. It takes effect when the application instance restarts. *index*: Platform CPU index. The platform CPU index is relative to a value of 10000 that is assigned to a configuration of a 1.0 GHz Celeron M CPU on the application runtime engine of network module NME_APPRE_302-K9. For example, the platform CPU index for the blade AIM_APPRE is 3000. The CPU utilization range varies between the minimum limit specified by the package to the maximum available by the system.

SSH Access to a Virtual Instance

Secure Shell (SSH) Protocol tunneling is a secure and effective solution for network users and administrators. SSH tunneling, also known as SSH port forwarding, is the process of forwarding selected TCP ports through an authenticated and encrypted tunnel.

This section consists of the following tasks:

- Configuring SSH Tunneling, page 63
- Configuring Shell Access in a Virtual Instance, page 64

For direct access to the Cisco AXP CLI, see the “Secure Shell Access to the Service Module” section on page 17.

Configuring SSH Tunneling

To configure SSH tunneling, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **ip ssh server** [*port-num*]

3. **ip ssh username** [**tunnel_root** | **tunnel_user**] **password** *clear-password-string*
4. **ip ssh username** [**tunnel_root** | **tunnel_user**] **password0** *clear-password-string*
5. **ip ssh username** [**tunnel_root** | **tunnel_user**] **password7** *hashed-password-string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 2	`ip ssh server [ port-num ]`	Starts or stops the SSH server on the specified port number. Port number range is 1–65535. Default port number is 22. Error messages: **Error Message** `Port is in use, please use another port.` This error occurs if the system cannot start the SSH server because the port is being used. Change the port number and restart. **Error Message** `Invalid port number, range is 1-65535` This error occurs if you use an invalid port number for the configuration.
Step 3	`ip ssh username [tunnel_root | tunnel_user] password` *clear-password-string*	Specifies a non encrypted (cleartext) user password. **tunnel_root**—Allows an SSH user with shell access to the application environment. **tunnel_user**—Allows an SSH user shell access to the application environment through a startup script that is implemented by the third party developer. The startup script decides on the level of access a user can have to perform specific operations.
Step 4	`ip ssh username [tunnel_root | tunnel_user] password 0` *clear-password-string*	Specifies a non encrypted password.
Step 5	`ip ssh username [tunnel_root | tunnel_user] password 7` *hashed-password-string*	Specifies a hidden password.

Configuring Shell Access in a Virtual Instance

To configure the SSH server for shell access in a virtual instance, perform the following steps.

Step 1 Install app_debug.pkg and **bind** the interface to ensure SSH connection.

```
SE-Module# configure terminal
config# app-service app-name
config-app-name# bind interface eth0
(config-app-name)# end
```

Step 2 Activate **tunnel_root** in the **ip ssh username [tunnel_root | tunnel_user] password** command by setting a password.

```
SE-Module# configure terminal
config# app-service app-name
config-myapp# ip ssh username tunnel_root password clear-password-string
config-myapp# end
```

Step 3 Enable the SSH server.

```
SE-Module# configure terminal
config# app-service app-name
config-app-name# ip ssh server
config-app-name# end
```

Step 4 Verify status of the SSH server.

```
SE-Module# app-service app-name
app-name# show ssh-server
```

Example:

```
Application SSH Server
Status: RUNNING
```

Step 5 You can now connect to the SSH server inside the virtual instance. By default, the SSH server port is 2022.

a. If you connect using **tunnel_root** as the SSH user, you have direct shell access after you log in with your password.

Note Use **tunnel_root** only in the development environment, for example when debugging.

```
Example:
workstation-shell# ssh tunnel_root@myblade -p 2022
tunnel_root@myblade's password:
vserver-shell#
```

b. If you connect using **tunnel_user** as the SSH user, the startup script in the third party application determines the type of access for a tunnel user.

Note Use **tunnel_user** only in the production environment.

In the example below, the startup script */usr/ssh/home/tunnel_user_app_startup.sh* does not allow a tunnel user to access the shell inside the virtual instance.

An interactive message appears after you log in with your password.

```
Example:
workstation-shell# ssh tunnel_user@myblade -p 2022
tunnel_user@myblade's password:
=====Start of message from the Cisco AXP Application SSH Support=====
Tunnel User (tunnel_user) has logged in successfully
Application specific Tunnel User startup script will be invoked (if exists)
=====End of message from the Cisco AXP Application SSH Support=======
Welcome Tunnel User!
Please enter 1 to do a 'pwd', or 2 to do a 'ls'
2
tunnel_user_app_startup.sh   tunnel_user_startup.sh
tunnel_user_app_startup.sh.sample
Connection to myblade closed.
```

```
Workstation-shell#
```

Configuring OSGi

Cisco AXP provides for the flexible management of Java applications through the OSGi framework. The OSGi framework defines a standardized, component-oriented, computing environment for networked services, and enables the remote and secure life cycle management of Java applications.

For more information on the OSGi specification, see the OSGi website at www.osgi.org.

To connect to OSGi, perform the following steps.

SUMMARY STEPS

1. **app-service** *application-name*
2. **connect osgi**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`connect osgi`	Cross-connects to the text console of the ProSyst OSGi framework. This command allows an administrator to manage the OSGi framework using the ProSyst commands. This command is only available when the ProSyst OSGi add-on packages are installed. For information about the latest version of the ProSyst OSGi add-on packages, see the *Cisco AXP Developer Guide* and the *Cisco AXP Release Notes*.
Step 3	`exit`	Exits application service mode.

Synchronizing Files

Cisco AXP provides developers with a data synchronization feature that allows them to synchronize files from a virtual instance to their workstation. The **sync file url** command synchronizes data from the virtual instance to the workstation using the rsync utility. The synchronization feature uses the rsync utility to exclude hard-linked files from the synchronization process.

Hard-linked files from the Guest-OS and other add-on files are excluded from synchronization since they are not packaged in an application. If an application requires a hard-linked file on the Cisco AXP service module to be overwritten with a file from a developer's workstation, it is necessary to first log into the Linux session in the virtual instance and remove the hard-linked file's protection.

If a file (or its directory) is deleted from one location and is used as the source of a synchronization operation, the file (or its directory) on the other location will not be automatically deleted. It must be manually deleted.

For example, if */work/helloworld-app-content/etc/mtab* is deleted from the workstation repository and **sync file url** command is invoked with the **in** keyword, the file on the Cisco AXP service module will not be automatically deleted.

Files that are not protected (not hard-linked from the Guest-OS or add-on files), will be synchronized out of the Cisco AXP service module for that virtual instance. These files include:

- All files added by the developer
- Temporary files used by run-time Linux
- Basic directory structures

The synchronization feature depends on:

- Guest-OS and add-on files hard-linked (UNIX file system link) in the virtual instance.
- rsync utility

Prerequisites

The following tasks must be performed before synchronizing files.

Workstation

On the workstation, the rsync utility is installed by default with the recommended Redhat platform. Cisco AXP invokes rsync remotely using SSH.

For the rsync utility to be initiated from the service module, the following tasks must be performed:

1. sshd must be running on the workstation
2. The workstation must be connected through the network from the router.
3. The developer must have a valid account on the workstation.
4. A directory must be available to contain the synchronization process content.
5. The ~/.ssh folder must have read, write and executable permissions for the owner.

Application

The application (empty application for starting developing) must be dependent on the application debug package to provide access to the Linux shell and rsync utility.

Since rsync requires network connectivity, the application must be configured to bind an interface.

Service Module

On the service module:

- Configure the **bind interface** command to connect the installed application to the workstation. Refer to the "Configuring External Network Interfaces" section on page 26.
- Configure SSH authentication keys to allow rsync session initiations without having to provide a password for each session. This configuration is required only once.

SUMMARY STEPS

1. **app-service** *application-name*

2. **sync file url rsync:***host_url* **direction [in|out] username** *username*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	a**pp-service** *application-name*	Enters application service mode.
Step 2	**sync file url rsync:***host_url* **direction [in|out] username** *username*	Wraps the rsync command. Before invoking the rsync utility, the command first identifies the files that can or should be synchronized. This identification process avoids synchronizing files that were hard-linked in the virtual instance, such as files from the Guest OS or other add-on files. **rsync**— Defines the rsync protocol *host_url*— Host URL **direction—**Direction of synchronizing the files: **in**—Content from the workstation is used as the master file. **out**—Content from the service module is used as the master file. *username*—Username used for authentication to the remote host (developer's workstation).
Step 3	**exit**	Exits application service mode.

Synchronization Example

This example assumes that the prerequisite tasks have been performed and the application is installed and configured on the Cisco AXP service module.

Setup

- Workstation (rsync repository) address: 192.168.1.4
- Username used by the developer on the workstation: john
- Empty application name: helloworld
- Cisco ISR prompt:

```
Router>
```

- Cisco AXP service module prompt:

```
appre>
```

- Workstation's folder to be used as a repository: /work/helloworld-app-content

Configuring SSH Authentication Keys

Configure the SSH authentication keys to establish trust between the service module and the workstation before using the **sync file url** command.

SUMMARY STEPS

1. Access a Linux shell session on the service module's application
2. Generate a key for the service module
3. Load the service module's public key to the workstation
4. Verify setup

Step 1 Setup a Linux session on the service module.

```
appre> app-service helloworld
appre>(exec-helloworld)> linux shell
bash-2.05b#
```

Step 2 Generate keys.

Use ssh-keygen to generate keys for the network-module.

Do not provide a passphrase. Save the generated keys under */root/.ssh/id_rsa* since ssh will be looking for it.

```
bash-2.05b# ssh-keygen -t rsa
```

a. Generate the public/private RSA key pair. You will enter information in interactive mode.

```
Enter the file in which you want to save the key (/root/.ssh/id_rsa): id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
```

The key fingerprint is:

```
d:31:68:8a:54:94:ee:9c:ba:14:79:41:53:ef:ac:ec root@appre
```

Step 3 Upload the service module's public key to the workstation to ensure that the service module's public key is known by the workstation. A simple way to upload the public key is to use the *scp* utility.

```
bash-2.05b# scp /root/.ssh/id_rsa.pub john@192.168.1.4:.ssh/authorized_keys2
```

Note *john* is the username or account on the workstation. We will use the same username to invoke the rsync command.

Step 4 Verify setup.

We can verify the setup of the authentication key by launching a simple SSH session from the service module to the workstation.

```
bash-2.05b# ssh john@192.168.1.4
```

You should not be prompted for a password for this command and should get a SSH session right away. If you do not initiate a SSH session immediately verify that:

- The right key has been uploaded. The key must be uploaded on the same machine to which you are initiating ssh during the verification step.
- The username is the same.
- The scp operation destination does not contain any typographical errors.

Step 5 Use the **sync file url** command

Perform data synchronization between the service module application files and the workstation repository. The first time you synchronize, it is recommended that you synchronize **out** (service module to the workstation) so that the base directory layout is exported.

Synchronizing **out** for the first time also makes it easier to add files to the structure.

In this example, we invoke the rsync command from the application context from the CLI prompt:

```
appre> app-service helloworld
appre(exec-helloworld)> sync file url rsync://192.168.1.4/work/helloworld-app-content
direction out username john
```

Using this command exports the data to the workstation 192.168.1.4 and places all the data of the application, which resides on the service module, under the workstation's directory */work/helloworld-app-content.*

The username john is used as credential for the connection.

The exported data must contain the following:

- Basic directory structure
- Temp files that were used in the virtual instance such has /tmp/*, /var/log/*, /var/lock/*
- Original files that were originally packaged in the application

At this point, files can be added and modified in the workstation repository and be synchronized back in, using the same command but using **in** instead of **out** as direction.

CLI Plug-in Invocation

The Cisco AXP CLI plug-in distribution service supports CLI plug-in actions in C, Java, and shell scripts.

Developers must implement APIs using the signatures provided in of the *Cisco AXP Developer Guide* and compile the APIs into application C libraries or Java classes. The CLI plug-in distribution service invokes these APIs when a user enters a command referring to one of these action classes.

EXEC Mode

Invoke an EXEC CLI plug-in as follows:

SUMMARY STEPS

1. **app-service** *application-name*
2. Enter the CLI plug-in as defined by the third party application.
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**app-service** *application-name* Example: SE-Module>(exec-helloworld) **disconnect users 20**	• *Invokes a plug-in command in EXEC mode.* • *application-name*— Application name. • Enter a query (?) to autogenerate a list of application names. • After entering the app-service sub-mode, enter the CLI plug-in defined by the third party application.
Step 2		Enters the CLI plug-in after entering app-service submode.
Step 3	**exit**	Exits application service custom mode.

Configuration Mode

Invoke a CLI plug-in using configuration mode as follows:

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application-name*
3. Enter the CLI plug-in defined by the third party application.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** SE_Module> **configure terminal**	Enters configuration mode.
Step 2	**app-service** *application-name*	• *Invokes a CLI plug-in using configuration mode.* • *application-name*— Application name. • Enter a query (?) to auto-generate a list of application names.
Step 3	Enter plugin CLI Example: SE-Module>(config-helloworld) **http ip 10.23.34.45 port 1234**	Enters the CLI plug-in after entering app-service submode.

Cisco IOS Service API Configuration

Cisco AXP provides service APIs to allow third parties to programmatically access, manage and augment existing Cisco IOS software networking features.

Common service APIs (supported in Bash, C, C++, Java, Perl, and Python) are as follows:

- Generic EXEC commands:

  This API allows an application to specify an EXEC command and return a string of output. The third party application must parse the output to retrieve the desired data.

  The supported EXEC CLIs are as follows:

  - **show** commands and/or their output modifiers.
  - **write memory** to save changes to NVRAM.
  - **copy running-config startup-config** to save changes to a startup configuration.
- Generic configuration commands:

  The service API allows an application to specify configuration commands consisting of a string of command(s) or a file path that contains commands separated with a delimiter such as a semicolon **;**

If a connection attempt fails to access a Cisco IOS NETCONF agent, the timeout value is set to 120 seconds. A FAIL code (1) is returned with an error message:

```
Fail to connect to IOS
```

It is the calling program's responsibility to allocate or free up the memory required to accommodate the response from Cisco IOS software for C programs. For Java and other languages, the default size of the response is 2048 bytes. A reply less than 2048 bytes is returned and embedded in the response string.

A reply larger than the size of the buffer allocated, or larger than the default value, the reply is placed in a file. A FILE code (2) is returned, and the file path and name are embedded in the response. The calling program can only retrieve the Cisco IOS software reply from the file.

Note *NETCONF over BEEP* must be enabled on the router and the Cisco AXP service module for this feature to work.

SUMMARY STEPS

Configure the following on the router:

1. **configure terminal**
2. **sasl profile** *profile-name*
3. **mechanism** *profile-mechanism*
4. **netconf max-sessions** *session-number*
5. **netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*] [**encrypt** *trustpoint*]

Configure the following on the Cisco AXP service module:

1. **netconf beep initiator** *router-IP-address port-number*
2. **netconf max-sessions** *session-number*

DETAILED STEPS

	Command or Action	Purpose
	Configure the following on the router:	
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**sasl profile** *profile-name*	Sets the SASL profile.
Step 3	**mechanism anonymous**	Configures the SASL profile mechanism.
Step 4	**netconf max-sessions** *num*	Specifies the maximum number of concurrent NETCONF sessions. *num*—Maximum number of sessions.
Step 5	**netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*] [**encrypt** *trustpoint*]	Specifies BEEP as the transport protocol for NETCONF and configures a peer as the BEEP listener.
	Configure the following on the Cisco AXP service module:	
Step 1	**netconf beep initiator** *router-IP-address port-number*	Specifies BEEP as the transport protocol for NETCONF sessions and configures a peer as the BEEP initiator.
Step 2	**netconf max-sessions** *num*	Specifies the maximum number of concurrent NETCONF sessions. *num*—Maximum number of sessions.

Verifying and Clearing Cisco IOS API Records

Use the **show history iosapi** and **clear history iosapi** commands to view EXEC and configuration command activities, and to clear specific records.

Use the **show log name messages.log** command in Cisco AXP EXEC mode to view the audit history.

Note The **show history iosapi** command helps to track command changes. You can view up to a hundred records where each record shows a configuration command change or an EXEC command change for a single virtual instance. Each virtual instance records up to seventy configuration command changes and thirty EXEC command changes.

SUMMARY STEPS

1. **app-service** *application-name*
2. **show history iosapi [***num* **]**
3. **show history iosapi** [**exec** | **config**] [*num*]
4. **clear history iosapi [***num* **]**
5. **clear history iosapi** [**exec** | **config**] [*num*]
6. **exit**
7. **show log name messages.log | include "iosapi audit"**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**app-service** *application-name*	Enters application service EXEC mode.
Step 2	**show history iosapi [***num***]**	Displays up to the last hundred records for configuration and EXEC modes. *num*—Number of records to be displayed.
Step 3	**show history iosapi** [**exec**|**config**][*num*]	Displays up to the last hundred records for the specified mode. **exec**—EXEC mode **config**—Configuration mode *num*—Number of records to be displayed.
Step 4	**clear history iosapi [***num***]**	Clears the number of specified records for configuration and EXEC modes. *num*—Number of records to be displayed.
Step 5	**clear history iosapi** [**exec**|**config**][*num*]	Clears the number of specified records for the specific mode. **exec**—EXEC mode **config**—Configuration mode *num*—Number of records to be cleared.
Step 6	**exit**	Exits application service EXEC mode.
Step 7	**show log name messages.log | include "iosapi audit"** `SE-Module> show log name messages.log|incude "iosapi audit"`	Displays audit history in Cisco AXP EXEC mode.

Cisco IOS Event Registration

To register an event using a third party application on Cisco AXP, either register an event using a configuration file or register an event using the CLI on the Cisco AXP service module. Then verify the events are registered. For further information, see the following sections:

- Registering an Event using the Event Configuration File, page 74
- Registering an Event using the Cisco AXP Service Module, page 75
- Verifying Registered Events, page 76

Registering an Event using the Event Configuration File

We recommend this registration method. You (the developer) register an event in an event configuration file in XML format. You then package the configuration file with the application and install the application package on the service module. See "Embedded Event Manager API" in the *Cisco AXP Developer Guide*.

Registering an Event using the Cisco AXP Service Module

To register an event on the Cisco AXP service module using the CLI, perform the following steps.

SUMMARY STEPS

1. **configure** *terminal*
2. **app-service** *application-name*
3. **event** *event-name* **register | unregister** *event-type*
4. **end**
5. **copy running-config startup-config**
6. **app-service** *application-name*
7. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters configuration mode.
Step 2	`app-service` *`application-name`*	Enters application service config mode.
Step 3	`event` *`event-name`* `register|unregister` *`event-type`*	Registers or unregisters an event and specifies the type of event. Example types: *cli*, *timer*, or *interface*. (For more event types, see "Embedded Event Manager API" in the *Cisco AXP Developer Guide*.)
Step 4	`end`	Exits config mode.
Step 5	`copy running-config startup-config`	Backs up the current configuration.
Step 6	`app-service` *`application-name`*	Enters app service config mode.
Step 7	`reset`	Resets the vserver for the application, so that the changes can take effect.

Verifying Registered Events

To verify the events registered for your application, perform the following step.

SUMMARY STEPS

1. **show run**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`show run`	Displays running configuration including registered events.
		Example:
		`show run`
		`app-service eemapi_test`
		`bind interface eth0`
		`event <event name> register <event type> <parameter>`
		Each line beginning with "event" shows an <event name> and <event type> that should correspond to one of the expected registered events for your application. If any events are missing, register the missing events using one of the two methods for "Cisco IOS Event Registration" section on page 74.
		`hostname se-10-1-2-7`
		`exit`

Cisco IOS Event Notification

The Cisco IOS software *Embedded Event Manager* (EEM) allows event detection and recovery on a Cisco IOS software device.

To be notified of a Cisco IOS event, you must:

1. Write and install an application using EEM APIs to receive events. For information on EEM APIs, refer to the "Embedded Event Manager API" section in the *Cisco AXP Developer Guide*.
2. Perform the following configuration steps for the router and Cisco AXP service module.

Note The configuration is similar to the "Cisco IOS Service API Configuration" section on page 71, except for steps to configure the **username**.

Configure the following on the router:

1. **configure terminal**
2. **username** *user-name* **password** *password* **privilege** *15*
3. **sasl profile** *profile-name*
4. **mechanism** *profile-mechanism*

5. **netconf max-sessions** *session-number*
6. **netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*]

Configure the following on the Cisco AXP service module:

1. **username ios** *user-name* **password** *password*
2. **netconf beep initiator** *router-IP-address port-number*
3. **netconf max-sessions** *session-number*

Using CLI to Trigger an EEM API Event

To manually trigger an ios_config event, perform the following steps.

1. **configure terminal**

   Enters configuration mode.
2. **ip host** *name ip-address*

   Configures the hostname and IP address.
3. **end**

   Exits configuration mode.

Using CLI to Trigger a Syslog Event

To manually trigger a syslog event by logging onto an interface, perform the following steps on the router.

1. **configure terminal**

   Enters configuration mode.
2. **logging buffered** *buffer-size*
3. **interface** *interface-name*

   Selects the type of interface to be recorded in the syslog. This is dependent upon having first set up an event-type of “syslog” with a pattern to be matched. For example, attribute pattern = “gigabitEthernet 0/1”.
4. **shutdown**

   Shuts down the interface. Sends event information to syslog.
5. **no shutdown**

   Starts the interface. Sends event information to syslog.
6. **exit**

   Exits configuration mode.

Troubleshooting a Cisco EEM API Configuration Event

To track ios_config events on the router, perform the following step.

- **debug beep session**

To track ios_config events on the Cisco AXP service module, perform the following step.

- **trace eemapi all**

To check the var/log/messages.log file perform the following step.

- **show log name messages.log**

In the following example, the ending of the command "**containing EEM paged**" causes the output to be filtered.

Example:

```
show log name messages.log containing EEM paged
<197>Nov 30 17:12:46 localhost EEMEventDaemon:    INFO EEMAPI DAEMON INFO
<197>Nov 30 17:12:46 localhost EEMEventDaemon: IOSConfigListener::receiveNotification
ENTER
<197>Nov 30 17:12:46 localhost EEMEventDaemon:    INFO EEMAPI DAEMON INFO
IOSConfigListener::receiveNotification: config change event received
<197>Nov 30 17:12:46 localhost EEMEventDaemon:    INFO EEMAPI DAEMON EVENT [eemapi_test]
event delivered: name=myiosevent, type=ios_config
<197>Nov 30 17:12:46 localhost EEMEventDaemon:    INFO EEMAPI DAEMON INFO
<197>Nov 30 17:12:46 localhost EEMEventDaemon: IOSConfigListener::receiveNotification:
finished processIOSConfigEvent
```

Modifying an Event

Use the **event** command to enable or disable the parameter of a registered event. Also use the **event** command to add, delete or modify an event. An event is not updated until a virtual instance reset command is issued.

SUMMARY STEPS

1. **configure terminal**
2. **app-service** *application name*
3. **[no] event** *event-name* {**register** | **unregister**} *event-type parameter*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters configuration mode.
Step 2	**app-service** *application name*	Enters application service mode.

Step 3	**[no] event** *event-name* {**register** | **unregister**} *event-type parameter*	Adds, edits, or deletes an event.
		register—Enables the event
		unregister—Disables the event
		`Example 1:`
		no event myevent
		Deletes the event myevent.
		`Example 2:`
		event mysecondevent **register** cli parameter1
		Adds the event mysecondevent (if it does not already exist), and enables parameter1.
		`Example 3:`
		event mysecondevent **unregister** cli parameter
		Disables parameter1 from the event mysecondevent.

Step 4	`exit`	Exits application service mode.

Application Status Monitor

Cisco AXP allows third party applications to plug-in their status monitoring and allows recovery from a malfunctioned state.

An application must provide one or more watchdog scripts or executable files bundled in their package to use the Cisco AXP application monitoring feature. The number of scripts or executables is dependent on the application, resulting in a unique way of determining the status of the application. For example, it can be based on Process Identifier (PID), or a response to an application ping. Cisco AXP supports Shell scripts and C language executables for application status monitoring.

For more information on watchdog scripts and executables, refer to the *Cisco AXP Developer Guide*.

The application status monitor has a heartbeat of 5 seconds, which is the minimum interval used for monitoring. For example, if the monitor interval is set at 12, monitoring of each virtual instance takes place every 12 heartbeat intervals, which is every one minute. You can configure the monitoring interval for a virtual instance through the **status-monitor monitor interval** command.

The scripts or executables return a status code where zero indicates that the application is healthy and alive. A non-zero status code indicates that the application is not functional. When a watchdog script or executable returns a non-zero status code, relevant information such as the name of the watchdog script, return status, and time of failure is logged.

A recovery counter counts the number of times the failure takes place, and acts like a delay mechanism for further action. A recovery count of three means that the application monitor has run for three iterations and is receiving either a non-zero return status, or the watchdog script has been running for over 3 monitoring intervals and is not returning a value.

You can use the **status-monitoring monitor interval** command for configuring the recovery threshold that decides on the number of recovery counters before taking the next action. When the recovery threshold is reached, the virtual instance restarts and the application monitor continues to run, repeating the monitoring cycle. A virtual instance can restart any number of times.

Third party developers can also provide default configuration parameters through a configuration file packaged together with their party application.

This section contains the following tasks:

- Configuring the Application Status Monitor Interval and Recovery Threshold, page 80
- Verifying the Application Status Monitor Output, page 80

Configuring the Application Status Monitor Interval and Recovery Threshold

To set the monitor interval and recovery threshold, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
1. **app-service** *application-name*
2. **status-monitor monitor_interval** *Interval-Num* **recovery_threshold** *Threshold-Num*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`configure terminal`	Enters configuration mode.
Step 2	`app-service` *`application-name`*	Enters application service mode.
Step 3	`status-monitor monitor_interval` *`Interval-Num`* `recovery_threshold` *`Threshold-Num`*	Configures the monitor interval and the recovery threshold. **monitor_interval**—Threshold value for monitoring interval. *Interval-Num*—Value is 1 to 99. Default is 12. Measured at 5 seconds per interval. **Recovery_threshold—**Threshold value for recovery attempts. *Threshold-Num*—Recovery threshold number from 1 to 99. Default is 5.
Step 4	`exit`	Exits application service mode.

Verifying the Application Status Monitor Output

To verify the status monitor output, perform the following step.

SUMMARY STEPS

1. **show-status monitor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`show-status monitor`	View the status monitor output. See the "Viewing the Application Status Monitor" section on page 85.

Verifying and Troubleshooting

This section consists of:

- Viewing Log and Core Files, page 81
- Clearing Log and Core Files, page 82
- Copying Files, page 83
- Syslog Server Logs, page 84
- Viewing the Application State, page 86
- Viewing Processes, page 87
- SSH Server Status, page 87
- Viewing Statistics, page 88
- Viewing Statistics, page 88
- Viewing Application Data, page 89
- Viewing Running Configuration, page 90
- Viewing System Resource Limits in Application Service Mode, page 90
- Viewing System Resource Limits in EXEC Mode, page 91
- Viewing a List of Installed Applications, page 91
- Resetting the Examples Application Environment, page 92
- Uninstalling an Application Package, page 92
- Common Troubleshooting Commands, page 93
- Verifying and Troubleshooting System Status, page 94
- Diagnostics and Logging Options, page 97

Viewing Log and Core Files

SUMMARY STEPS

1. **app-service** *application-name*
2. **show cores**
3. **show logs**

4. **show log name** l*og-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`show cores`	Lists core files that reside in the application service environment.
Step 3	`show logs`	Displays all the log files under /var/log directory of the virtual instance.
Step 4	`show log name` *`log-name`*	Displays the specified log. Keyword options are: **containing**—Only display events matching a regex pattern. **page** —Displays in page mode. **tail**—Waits for events and prints them as they occur. | Pipe output to another command

Clearing Log and Core Files

Log files can also be cleared in system EXEC mode.

SUMMARY STEPS

1. **app-service** *application-name*
2. **clear cores**
3. **clear logs**
4. **clear core** *core-name*
5. **clear log** *log-name*
6. **exit**

System EXEC mode

7. **clear logs**
8. **clear log name** *log-name*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *application-name*	Enters application service mode.
Step 2	`clear cores`	Clears all core files of the application.
Step 3	`clear logs`	Clears content of all log files of the application.
Step 4	`clear core name` *core-name*	Clears the specified core file of the application.
Step 5	`clear log name` *log-name*	Clears the content of the specified log file of the application.
Step 6	`exit`	Exits application service mode.
	In system EXEC mode:	**Note** System EXEC mode is similar to Privileged EXEC mode in Cisco IOS software.
Step 7	`clear logs`	Clears contents of all host log files except the syslog server log files.
Step 8	`clear log name` *log-name*	Clears contents of the specified host log file. This command does not clear a syslog server log file.
Step 9	`exit`	Exits system EXEC mode.

Copying Files

Core names and log names can contain wildcards *. You can copy log files in application service EXEC sub-mode or in system EXEC mode.

SUMMARY STEPS

1. **app-service** *application-name*
2. **copy core** *core-name* **url** *ftp/http url*
3. **copy log** *log-name* **url** *ftp/http url*
4. **copy logs bundle** *destfilename.tar* **url** *url*
5. **exit**

In system EXEC mode:

6. **copy log** *log-name* **url** *ftp/http url*
7. **copy logs bundle** *destfilename.tar* **url** *url*
8. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`copy core` *`core-name`* `url` *`ftp/http url`*	Copies the specified core file to a remote URL. *ftp/http url*— FTP or HTTP server address The standard ftp URL format is supported: `ftp://[user-id:ftp-password@]ftp-server-address[/directory]`
Step 3	`copy log` *`log-name`* `url` *`ftp/http url`*	Copies syslog, trace and custom application log files for the specified application to a remote URL. The Log name may contain wildcards * .
Step 4	`copy logs bundle` *`destfilename.tar`* `url` *`url`*	Copies a tar file containing syslog files, and custom application log files from the guest operating system to a remote URL. *destfilename.tar*— Tar filename *url*—Destination URL
Step 5	`exit`	Exits application service mode.
	In system EXEC mode:	System EXEC mode is similar to Privileged EXEC mode in Cisco IOS software.

Step 6	`copy log` *`log-name`* `url` *`ftp/http url`*
	`Example:`
	`Se-Module>` **`copy log`** `log-name url ftp/http url`
		Copies Cisco AXP host operating system log files to a remote URL.
		Wildcards * may be used to copy more than one log file at a time.

Step 7	`copy logs bundle` *`destfilename.tar`* `url` *`url`*
	`Example:`
	`SE-Module>` **`copy logs bundle`** *`destfilename.tar`* **`url`** *`url`*
		Copies a tar file containing syslog files, and custom application log files on the host and guest operating sytems to a remote URL.
		This command does not copy the remote syslog server log files.
		destfilename.tar— Tar filename
		url—Destination URL

Step 8	`exit`	Exits system EXEC mode.

Syslog Server Logs

All commands are in system EXEC mode.

SUMMARY STEPS

1. **show syslog-server logs**
2. **show syslog-server log name** *log-name*
3. **clear syslog-server logs**
4. **clear syslog-server log name** *log-name*

5. **copy syslog-server logs bundle** *destination-filename.gz* **url** *ftp/http url*
6. **copy syslog-server log name** *log-name* **url** *ftp/http url*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`show syslog-server logs`	Lists all the syslog server log files.
Step 2	`show syslog-server log name` *`log-name`*	Displays the specified syslog server log file. Keyword options are: **containing**— Specify regexp to filter **paged**—Display in page mode **tail**—Wait for events and print them as they occur | — Pipe output to another command
Step 3	`clear syslog-server logs`	Clears the contents of all the syslog server log files.
Step 4	`clear syslog-server log name` *`log-name`*	Clears contents of the specified syslog server log file.
Step 5	`copy syslog-server logs bundle` *`destination-filename.gz`* `url` *`ftp/http url`*	Bundles all the syslog server log files into a gzip file and copy it to a remote URL. *destination-filename.gz*—gzip filename *ftp/http url*—Destination URL
Step 6	`copy syslog-server log name` *`log-name`* `url` *`ftp/http url`*	Copies the specified syslog server log file. Wildcard * may be used to copy more than one log file at a time.
Step 7	`exit`	Exits system EXEC mode.

Viewing the Application Status Monitor

SUMMARY STEPS

1. **app-service** *application-name*
2. **show status-monitor**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	a**pp-service** *application-name*	Enters application service mode.

Step 2	**show status-monitor**	Displays parameters of the status monitor.
		The monitor status shown in the example can have one of the following values:
		--- : Monitor has not been turned ON.
		PASSED: Monitoring reports successful execution of watchdog scripts.
		RECOVERY: Monitoring reports a watchdog failure, or the watchdog is taking longer than the monitor interval to return a value. The virtual instance restarts if the recovery threshold period is exceeded.
		Example:
		`Application: helloworld`
		`Monitor status: PASSED`
		`Monitor in progress: Yes`
		`Last executed watchdog: W00template.sh`
		`Last executed date: Wed Sep  5 14:09:58 PDT 2007`
		`Last failed watchdog: ---`
		`Last failed return code: -`
		`Last failed date: ---`
		`Last restarted date: ---`
		`Recovery threshold: 4`
		`Monitor interval: 3`

Step 3	**exit**	Exits application service mode.

Viewing the Application State

SUMMARY STEPS

1. **app-service** *application-name*
2. **show state**
3. **show state details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	a**pp-service** *application-name*	Enters application service mode.
Step 2	**show state**	Displays the state and health of the specified application as: State—Online, Offline, Pending-online, Pending-offline. Health—Alive or Down.
Step 3	**show state details**	Displays status related information of the virtual instance.

Viewing Processes

SUMMARY STEPS

1. **app-service** *application-name*
2. **show process**
3. **show process running**
4. **show process all**
5. **show process pid** *process-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**app-service** application-name	Enters application service mode.
Step 2	**show process**	Shows all processes running in the application environment sorted by process ID in ascending order.
Step 3	**show process running**	Shows all running processes in the application environment sorted by CPU usage in descending order.
Step 4	**show process all**	Shows all processes running in the application environment with a summary of CPU and memory tasks in the application environment.
Step 5	**show process pid** *process-id*	Shows the process, specified by the process ID, running in the application environment.

SSH Server Status

SUMMARY STEPS

1. **app-service** *application-name*
2. **show ssh-server**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *application-name*	Enters application service mode.
Step 2	`show ssh-server`	Displays the current status of the SSH server.

Viewing Statistics

SUMMARY STEPS

1. **app-service** *application-name*
2. **show statistics**
3. **show statistics app**
4. **end**
5. **show app-service statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`app-service`** *`application-name`*	Enters application service mode.

Step 2	**`show statistics`**	Displays statistics such as CPU utilization and memory for a virtual instance in the application environment.
		Example:
		`CTX PROC  VSZ   RSS   userTIME  sysTIME  UPTIME  NAME`
		`2    3    6.6M  2.5M  0m00s12   0m00s40  3h04m08 Test1`
		
		`CTX = context number for the virtual instance`
		`PROC = quantity of processes in the context`
		`VSZ = number of pages of virtual memory`
		`RSS = Resident set size limits for memory`
		`userTime = utime User-mode CPU time accumulated`
		`sysTime = ctime Kernel-mode CPU time accumulated`
		`upTime = uptime`

Step 3	**`show statistics app`**	Displays statistics of third party applications integrated into the application environment. When you use this command, */bin/appstats* is executed. The third party application must provide the *appstats* file, in binary or script format, to plug in for its statistics.
Step 4	`end`	Exits application service mode.

Step 5	**`show app-service statistics`**
	`Example:`
	`Se-Module> show app-service statistics`
		(System EXEC mode) Lists all the installed virtual instances and applications, and displays the application instance's memory and processing time information.

Viewing Application Data

SUMMARY STEPS

1. **app-service** *application-name*
2. **show tech-support**
3. show tech-support details
4. **exit**

In System EXEC mode:

5. **show tech-support details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`show tech-support`	Dumps information on the terminal provided by the third party application. Displays running-config, state, resource limits, and statistics of the application environment. Executes the **/bin/techsupport** binary or script file to display application-specific information if provided by the third party application.
Step 3	`show tech-support details`	Displays detailed technical support information for the third party application. This command displays output from the show commands applied to the third party application.
Step 4	`exit`	Exits application service mode.
	In System EXEC mode:	**Note** System EXEC mode is similar to Privileged EXEC mode in Cisco IOS software.
Step 5	`show tech-support details`	Displays additional status information related to the virtual instance.

Viewing Running Configuration

SUMMARY STEPS

1. **app-service** *application-name*
2. **show running-configuration**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`show running-configuration`	Displays running configuration only for the application environment.
Step 3	`exit`	Exits application service mode.

Viewing System Resource Limits in Application Service Mode

To view system resource limits in application service mode, perform the following steps.

SUMMARY STEPS

1. **app-service** *application-name*
2. **show resource limits**

3. show resource limits details

DETAILED STEPS

	Command or Action	Purpose
Step 1	**app-service** application-name	Enters application service mode.

Step 2	**show resource limits**	Displays the settings of the system resource limits for the application environment.
		`Example:`
		`APPLICATION CPU(INDEX) MEMORY(KB) DISK(KB) LOG(MB)`
		`Guestos1     7000       2000      10000    50`

Step 3	**show resource limits details**	Displays raw information for a virtual instance.

Viewing System Resource Limits in EXEC Mode

To view system resource limits in EXEC mode, perform the following steps.

SUMMARY STEPS

1. **show resource limits**
2. **show resource limits details**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show resource limits**	Displays CPU, memory, disk, and other system resource limits set for: • Host operating system • Each installed application service
Step 1	**show resource limits details**	Displays runtime statistical information for all virtual instances.

Viewing a List of Installed Applications

SUMMARY STEPS

1. **show app-service state (system EXEC mode)**
2. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`show app-service state`
	`Example:`
	`SE-module> show app-service state`
		Displays a list of the installed applications and their state and health in system EXEC mode.
		System EXEC mode is similar to privileged EXEC mode in Cisco IOS software.
		State— Online or offline. Indicates if the virtual environment is running.
		Health—Alive or down.
		Health refers to the status of the internal application.
		This status is communicated back to the Cisco AXP environment through an API call by the application monitoring process.

Step 2	`exit`	Exits application service mode.

Resetting the Examples Application Environment

SUMMARY STEPS

1. **app-service** *application-name*
2. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`app-service` *`application-name`*	Enters application service mode.
Step 2	`reset`	Resets the specified application environment and places it in the current state. For example: • It forces the virtual environment to stop if it is currently in the shutdown state (offline or pending-offline). • If the virtual environment is currently online, it forces a shutdown and restarts the virtual environment to bring it back online again.

Uninstalling an Application Package

To uninstall an application package, perform the following step.

SUMMARY STEPS

1. **software uninstall** *package name* (system EXEC mode)

DETAILED STEPS

	Command or Action	Purpose
Step 1	`SE-Module> software uninstall package name`	Uninstalls the previously installed application software package. The uninstallation will not work if the package does not exist, or if other packages depend on the specified package being uninstalled.

Common Troubleshooting Commands

Tables 3 and 4 show some common router and service module commands.

To view a complete list of available commands, type **?** at the prompt
Example: `Router(config-if)# ?`.

To view a complete list of command keyword options, type **?** at the end of the command
Example: `Router# service-module integrated-service-engine?`.

Table 3 on page 93 and Table 4 on page 94 group commands by the configuration mode in which they are available. If the same command is available in more than one mode, it may act differently in each mode.

To shut down or start up the service module or the Cisco AXP software application that runs on the module, use shutdown and startup commands as needed from Table 3.

Note

- Some shutdown commands can potentially disrupt service. If command output for such a command displays a confirmation prompt, confirm by pressing **Enter** or cancel by typing **n** and pressing **Enter**. Alternatively, prevent the prompt from displaying by using the **no-confirm** keyword.
- Some shutdown commands shut the module or application down and then immediately restart it.

Table 3 Common Shutdown and Startup Commands

Configuration Mode	Command	Purpose
Router#	**service-module integrated-service-engine** *slot*/**0 reload**	Shuts down the service-module operating system gracefully, then restarts it from the bootloader.
Router#	**service-module integrated-service-engine** *slot*/**0 reset**	**Caution** Use this command with caution. It does *not* provide an orderly software shutdown and consequently may impact file operations that are in progress. Resets the hardware on a module. Use only to recover from shutdown or a failed state.

Table 3 Common Shutdown and Startup Commands

Configuration Mode	Command	Purpose
`Router#`	**service-module integrated-service-engine** *slot*/**0 session**	Accesses the specified service engine and begins a service-module configuration session.
`Router#`	**service-module integrated-service-engine** *slot*/**0 shutdown**	Shuts down the service-module operating system gracefully. Use when removing or replacing a hot-swappable module during online insertion and removal (OIR).
`Router#`	**service-module integrated-service-engine** *slot*/**0 status**	Displays configuration and status information for the service-module hardware and software.
`Router(config)#`	**shutdown**	Shuts down the entire system (host router plus service module) gracefully.
`SE-Module bootloader>`	**boot**	Starts the bootloader, boothelper, or application.
`SE-Module(offline)>`	**reload**	Performs a graceful halt and reboot of a service-module operating system.
`SE-Module>`	**reboot**	Shuts down Cisco AXP software without first saving configuration changes, then reboots it from the bootloader.
`SE-Module>`	**reload**	Shuts down Cisco AXP software gracefully, then reboots it from the bootloader.
`SE-Module>`	**shutdown**	Shuts down the Cisco AXP software application gracefully, then shuts down the module.

Verifying and Troubleshooting System Status

To verify the status of an installation, upgrade, downgrade, or troubleshoot problems, use verification and troubleshooting commands as needed from Table 4.

Note Keyword options for many **show** commands include provision to display diagnostic output on your screen or pipe it to a file or a URL.

Table 4 Common Verification and Troubleshooting Commands

Configuration Mode	Command	Purpose
`Router#`	**ping**	Pings a specified IP address to check network connectivity (does not accept a hostname as destination).
`Router#`	**show arp**	Displays the current Address Resolution Protocol (ARP) table.

Table 4 Common Verification and Troubleshooting Commands (continued)

Configuration Mode	Command	Purpose
`Router#`	**show clock**	Displays the current date and time.
`Router#`	**show configuration**	Displays the current bootloader configuration as entered using the **configure** command.
`Router#`	**show controllers integrated-service-engine**	Displays interface debug information.
`Router#`	**show diag**	Displays standard Cisco IOS software diagnostics information, including information about Cisco AXP software.
`Router#`	**show hardware**	Displays information about service-module and host-router hardware.
`Router#`	**show hosts**	Displays the default domain name, style of name lookup, list of name-server hosts, and cached list of hostnames and addresses.
`Router#`	**show interfaces**	Displays information about all hardware interfaces, including network and disk.
`Router#`	**show ntp status**	Displays information about Network Time Protocol (NTP).
`Router#`	**show processes**	Displays a list of the running application processes. Displays the state of the sshd process.
`Router#`	show processes memory	Displays information about the sshd process when it is running.
`Router#`	**show running-config**	Displays the configuration commands that are in effect.
`Router#`	**show startup-config**	Displays the startup configuration.
`Router#`	**show tech-support**	Displays general information about the host router that is useful to Cisco technical support for problem diagnosis.
`Router#`	**show version**	Displays information about the loaded router-software or service-module-bootloader version and also hardware and device information.
`Router#`	**test scp ping**	Pings the service module to check network connectivity.
`Router#`	**verify**	Displays version information for installed hardware and software.
`SE-Module>`	**ping**	Pings a specified IP address to check network connectivity (does not accept a hostname as destination).
`SE-Module>`	**show arp**	Displays the current Address Resolution Protocol (ARP) table.

Table 4 Common Verification and Troubleshooting Commands (continued)

Configuration Mode	Command	Purpose
`SE-Module>`	**show clock**	Displays the current date and time.
`SE-Module>`	**show config**	Displays the current bootloader configuration as entered by the **configure** command.
`SE-Module>`	**show hosts**	Displays the default IP domain name, lookup style, name servers, and host table.
`SE-Module>`	**show interfaces**	Displays information about the service-module interfaces.
`SE-Module>`	**show interface GigabitEthernet**	Displays basic interface configuration information about an Ethernet interface.
`SE-Module>`	**show ntp status**	Displays information about Network Time Protocol (NTP).
`SE-Module>`	**show processes**	Displays a list of the running application processes.
`SE-Module>`	**show running-config**	Displays the configuration commands that are in effect.
`SE-Module>`	**show software directory download**	Displays the contents of the downgrade or download directory on the download FTP file server.
`SE-Module>`	**show software download server**	Displays the name and IP address of the configured download FTP file server.
`SE-Module>`	**show software licenses**	Displays license information for installed packages.
`SE-Module>`	**show software packages**	Displays version information for installed packages.
`SE-Module>`	**show software versions**	Displays version information for installed software.
`SE-Module>`	**show startup-config**	Displays the startup configuration.
`SE-Module>`	**show syslog-server**	Displays syslog server status.
`SE-Module>`	**show tech-support**	Displays general information about the service module that is useful to Cisco technical support for problem diagnosis.
`SE-Module>`	**show trace**	Displays the contents of the trace buffer.
`SE-Module>`	**show version**	Displays information about the loaded router-software or service-module-bootloader version and also hardware and device information.
`SE-Module>`	**software remove**	Removes all files, downloaded package and payloads, or stored downgrade files created during an upgrade.

Diagnostics and Logging Options

To configure logging options for Cisco AXP software, use logging commands from Table 5.

Note Keyword options for many **log** and **trace** commands include provision to display diagnostic output on your screen or to pipe it to a file or a URL.

Table 5 Common Logging Commands

Configuration Mode	Command	Purpose
`SE-Module>`	**log console monitor**	Configures error logging by means of console logging (logged messages are displayed on the console).
`SE-Module(config)>`	**log server**	Configures error logging by means of a system-log (syslog) server (syslog is an industry-standard protocol for capturing log information for devices on a network).

Diagnostics consist of two types:

- System log (syslog)—Syslog is an industry standard protocol for capturing the following events:
  - Fatal exceptions that cause an application or system crash, during which normal error-handling paths are typically nonfunctional
  - Application run-time errors that cause unusual conditions and configuration changes

  The syslog file size is fixed at (AIM) 1 MB or (NM) 10 MB. Syslog configurations survive a power failure.
- Traces—Trace logs capture events related to the progress of a request through the system.

  Trace logs survive a CPU reset; trace configurations survive a power failure. Log and display these with the **trace** commands.

To generate and display syslog and trace diagnostics, use trace commands as needed from Table 6.

Table 6 Common Trace Commands

Configuration Mode	Command	Purpose
`SE-Module>`	**clear trace**	Clears logged trace events for specified modules.
`SE-Module>`	**log trace**	Logs configured traces to the service module (can be done locally or remotely).
`SE-Module>`	**no trace**	Disables tracing for specified modules, entities, or activities.
`SE-Module>`	**show errors**	Displays error statistics by module, entity, or activity.
`SE-Module>`	**show trace**	Displays trace settings.
`SE-Module>`	**show trace buffer**	Displays the contents of the trace buffer.

Table 6 Common Trace Commands (continued)

Configuration Mode	Command	Purpose
`SE-Module>`	**show trace store**	Displays the contents of the traced messages that are stored.
`SE-Module>`	**trace**	Enables tracing (that is, generates error reports) for specified modules, entities, or activities.

Configuring the Bootloader

The router must be configured correctly before setting up the bootloader. The service module will not connect to the external network if the router is not configured correctly.

To configure the bootloader, perform the following steps.

Step 1 Enter the following commands from the router CLI:

a. **service-module Service-Engine 1/0 reset** (wait about 10 seconds after issuing this command)

b. **service-module Service-Engine 1/0 session** (if the first try fails, repeat this command)

Step 2 Wait for the following prompt:

“Please enter '***' to change boot configuration”.

a. Enter “***” to drop the service module into the bootloader.

Step 3 Enter the **config** command to configure the bootloader:

SE- boot-loader> **config**

a. Enter these parameters:

IP address —Service module IP address as configured on the host router

Subnet mask— Service module subnet mask as configured on the host router

TFTP server—(Optional) IP address of the TFTP server with the helper image

Gateway—Gateway address as configured in the host router

Default helper-file—(Optional) Filename of the helper image

Ethernet interface—**Internal**

Note The internal interface is facing the router. The external interface may or may not be present on the service module.

Step 4 **Default Boot**: **disk**

Step 5 **Default bootloader: secondary**

Note Always use the secondary bootloader; primary is only for backup.

Step 6 To enter boot helper type: **boot helper**

or

To boot normally from the disk type: **boot disk**

Tip Use the boot helper to reboot with a helper image if the service module does not boot up with the regular image. See the "Installing Software using a Helper Image" section on page 16.

Additional References

This section consists of the following topics.

- Related Documentation, page 99

Related Documentation

- *Cisco 1800 Series Integrated Services Routers*
- *Cisco 3800 Series Integrated Services Routers*
- *Installing and Upgrading Internal Modules in Cisco 1800 Series Routers (Modular)*
- *Cisco IOS Security Configuration Guide, Release 12.4*
- *NETCONF over BEEP*
- *Configuring VRF Lite*
- *Virtual LANs/VLAN Trunking Protocols*
- *Router IP Traffic Export (RITE)*
- *RITE Packet Capture Enhancements*
- *Cisco Network Analysis Module Software*
- *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW*
- *Cisco Branch Routers Series Network Analysis Module*
- *OSGi Framework*