



Cisco Application eXtension Platform Quick Start Guide

Last Revised: 7/10/08

This guide describes the tasks and commands necessary to quickly configure the hardware and software for the Cisco Application eXtension Platform (Cisco AXP). The hardware is any Cisco 2800 series Integrated Services Router (called Cisco 280x ISR hereafter) and a Cisco AXP service module. This guide also describes how to install and configure third-party applications on Cisco AXP.

For details on configuration tasks and developing third-party applications, see the following two guides:

- [Cisco AXP Developer Guide](#)
- [Cisco AXP User Guide](#)

This guide contains the following sections:

- [Cisco AXP Documentation Roadmap, page 1](#)
- [Syntax Conventions, page 3](#)
- [Overview, page 3](#)
- [Hardware Installation, page 3](#)
- [Configuring the Router, page 4](#)
- [Configuring the Cisco AXP Service Module, page 5](#)
- [Packaging Applications on Cisco AXP, page 8](#)
- [Installing and Uninstalling an Application, page 14](#)
- [Viewing the Application Environment, page 17](#)
- [Configuring Network Services, page 17](#)
- [Where to Go Next, page 19](#)
- [Appendix A: Cisco 280x ISR and Cisco AXP Service Module Configurations, page 20](#)
- [Notices, page 22](#)

Cisco AXP Documentation Roadmap

For getting started quickly, we recommend that you start by reading this guide, and next read the two other guides: the [Cisco AXP Developer Guide](#) and the [Cisco AXP User Guide](#).

The *Cisco AXP User Guide* explains the configuration of the Cisco 280x ISR and the Cisco AXP service module (blade).

The Cisco AXP Developer Guide describes how to create, package and install applications.

Links to Cisco AXP and related documentation are shown in [Table 1](#).

Table 1 *Cisco AXP Documentation*

Description	Title	Description
Quick Start guide	<i>Cisco AXP Quick Start Guide</i>	Getting started with a minimal configuration for Cisco AXP.
User guide	<i>Cisco AXP User Guide</i>	Configuration of Cisco 280x ISR and Cisco AXP service module.
Developer guide	<i>Cisco AXP Developer Guide</i>	Creation, packaging, and installation of applications on Cisco AXP.
Release notes	<i>Release Notes for Cisco AXP 1.0.1</i>	Supported hardware, downloadable files, and caveats for a particular version of Cisco AXP.
Command reference	<i>Cisco AXP 1.0 Command Reference</i>	Software commands relating to Cisco AXP.
Modules hardware reference guides	<i>Cisco Network Modules: Install and Upgrade Guides</i>	Cisco network module installation guide.
Router hardware reference guides	<i>Installing and Upgrading Internal Modules in Cisco 2800 Series Routers</i>	Includes “Installing Network Modules in Cisco 2800 Series Routers” section.

Syntax Conventions

Table 2 on page 3 shows the command syntax conventions used in this guide.

Table 2 Command Syntax Conventions

Convention	Description	Examples
Courier Bold	Commands, keywords, and arguments/parameters.	<code>configure terminal</code>
^ or <i>Ctrl</i>	Keys labeled <i>Control</i> .	<code>^D</code> or <i>Ctrl-D</i> indicates the action of holding down the Control key while pressing the D key
Courier Plain	Output, displayed on the screen.	<code>[root@fedora tools]</code>
< >	<ul style="list-style-type: none"> passwords. Key to press. 	<code><enter></code>
[]	<ul style="list-style-type: none"> Default text displayed by system. optional keywords or arguments. 	<pre>Building configuration... [OK] rpm_extractor.sh --proj <project-directory> [--output] [--scripts] [--deps] <RPM files></pre>
<i>italics</i>	<ul style="list-style-type: none"> Nonprinted characters, such as a password. Placeholder for file directory name, or list of filenames. 	<pre># ISR-enable-password <enter> rpm_extractor.sh --proj project-directory [--output] [--scripts] [--deps] rpm-files</pre>

Overview

The purpose of this guide is to provide steps and examples for you to install, configure, and get started with application development on Cisco AXP.

For an overview of Cisco AXP, see the “Cisco Application eXtension Platform Overview” section in the [Cisco AXP Developer Guide](#).

The example-based tasks in this guide are based on running configuration files shown in “Appendix A: Cisco 280x ISR and Cisco AXP Service Module Configurations” section on page 20.

Hardware Installation

Prerequisites

Before you install the Cisco 280x ISR and Cisco AXP service module, check the hardware compatibility matrix to make sure you have the correct Cisco 280x platform and service module. For a list of the latest Cisco AXP ISR models and service module SKUs, see the [Release Notes for Cisco AXP 1.0.1](#).

Installation

To install the Cisco 280x ISR and Cisco AXP service module, see the platform-specific hardware installation and configuration guides available on Cisco.com. For example, see the hardware guides listed in [Table 1 on page 2](#).

Configuring the Router

To configure the Cisco 280x ISR, perform the following steps.

Step 1 Open a Telnet session.

For the telnet session, use one of the following methods:

- DOS window
 - Secure shell
 - Telnet/SSH client emulation software such as PuTTY
- a. Specify the IP address of the Cisco 280x ISR.

```
telnet ip-address
```

b. Type your username and password for the Cisco 280x ISR.

```
Example:
telnet 192.168.1.39 <enter>
Trying 192.168.1.39 ... Open
....

User Access Verification

Username: cisco <enter>
Password: cisco <enter>
2800_w_axp#
```

c. The # sign signifies that you are in enable mode, which allows you more privileges in the Cisco 280x ISR environment.

d. If you see a > sign instead of the # sign, type the **enable** command, followed by the *ISR-enable-password*:

```
2800_w_axp> enable <enter>
Password: ISR-enable-password <enter>
2800_w_axp#
```

Step 2 Verify the running configuration.

Check the running configuration using the **show running-config** command. The output shown here is from the running configuration shown in the [“Appendix A: Cisco 280x ISR and Cisco AXP Service Module Configurations”](#) section on page 20.

```
2800_axp# show running-config
!
!
interface FastEthernet0/0
 shutdown
 no mop enabled
!
interface FastEthernet0/1
 ip address 192.168.1.39 255.255.255.0
```

```

duplex full
speed auto
no mop enabled
!
interface Integrated-Service-Engine 1/0
shutdown
no keepalive
no mop enabled
!
!
```

**Tip**

Press the space bar to scroll through the output if the entire configuration is not visible on your screen.

The running configuration shows multiple interfaces, for example: FastEthernet0/0 (FE0/0), FastEthernet0/1 (FE0/1), and Integrated-Service-Engine1/0 (ISE1/0). The number of interfaces on your Cisco 280x ISR depends on the model and number of service modules that are installed. See [Table 1 on page 2](#) for links to hardware documentation.

- Step 3** Use the **show interfaces** command to view all the interfaces, or use the command with keywords to view specific interfaces:

```

2800_w_axp# show interfaces FastEthernet 0/0
2800_w_axp# show interfaces FastEthernet 0/1
2800_w_axp# show interfaces Integrated-Service-Engine 1/0
```

**Tip**

Press the space bar repeatedly to scroll through the output.

Configuring the Cisco AXP Service Module

This section consists of:

- [Cisco AXP Command Modes, page 5](#)
- [Configuring the Cisco AXP Service Module Interface, page 6](#)

Cisco AXP Command Modes

The Cisco AXP command environment on the service module consists of the command modes shown in [Table 3](#).

Table 3 Cisco AXP Command Modes

Mode	Example
Cisco AXP application service configuration mode	se-Module(config)> app-service helloworld se-Module(config-helloworld)>
Cisco AXP application service EXEC mode	se-Module> app-service application-name Example: se-Module> app-service helloworld se-Module(exec-helloworld)>

Table 3 Cisco AXP Command Modes

Mode	Example
Cisco AXP configuration mode	se-Module> config t se-Module(config)>
Cisco AXP EXEC mode ¹	se-Module>

1. This mode is similar to the Cisco IOS privileged EXEC mode.

**Note**

To exit the Cisco AXP command environment and return to the Cisco 280x ISR command environment:

1. Return to Cisco AXP EXEC mode, and
2. Type **exit**.

For example:

```
se-Module> exit
2800__axp#
```

Configuring the Cisco AXP Service Module Interface

Configure the Integrated Service Engine interface from the Cisco 280x ISR, using the Cisco IOS CLI.

The ISE1/0 port connects the ISE1/0 service module to the router. The ISE1/0 port is internal and you cannot physically see it by looking at the Cisco 280x ISR. (In the examples in this document, the service module is “se-Module”, see examples in [Table 3](#)).

In order to gain IP connectivity to the Cisco AXP service module, you must first configure IP parameters from the Cisco 280x ISR. You must also give the Cisco AXP service module a default gateway that it can use to ensure full connectivity of your application.

To configure the Cisco AXP Service Module, perform the following steps:

Step 1 Enter configuration mode, by entering the following command:

```
2800_w_axp# configure terminal <enter>
2800_w_axp(config)#
```

You can now make changes to the current running configuration file. For a complete running configuration used in the examples in this guide, see the [“Appendix A: Cisco 280x ISR and Cisco AXP Service Module Configurations”](#) section on page 20.

Step 2 To configure the ISE1/0 interface, enter the configuration context for the interface, by typing the following command:

```
2800_w_axp(config)# interface Integrated-Service-Engine 1/0 <enter>
2800_w_axp(config-if)#
```

Step 3 Check for basic IP connectivity by entering the following three commands.

**Note**

In this example the segment, which is represented by the gigabit Ethernet connection between the Cisco 280x ISR and Cisco AXP service module, must be given its own routable IP subnet. This IP subnet is different from the IP subnet used for the Cisco 280x ISR FE0/0 interface.

- a. Assign an IP address to the Cisco 280x ISR side of the ISE1/0 connection.
Type the following:

```
2800_w_axp(config-if)# ip address 192.168.2.1 255.255.255.0
```

- b. Assign an IP address to the Cisco AXP service module side of the ISE1/0 connection.
Type the following:

```
2800_w_axp(config-if)# service-module ip address 192.168.2.2 255.255.255.0
```

- c. Assign an IP default gateway that your Cisco AXP service module can use.
Type the following:

```
2800_w_axp(config-if)# service-module ip default-gateway 192.168.2.1
```

The Cisco AXP service module IP default gateway address is the Cisco 280x ISR interface. The Cisco AXP service module sends outgoing traffic through this default interface to the Cisco 280x ISR using an internal gigabit Ethernet connection.

- Step 4** Enable the ISE1/0 interface by taking it out of its default state of shutdown. This is an administrative function that keeps an interface down so that no IP packets are transmitted or received. Type the following:

```
2800_w_axp(config-if)# no shutdown
```

From the Cisco 280x ISR perspective, your Cisco AXP service module (ISE1/0) is now connected.

- Step 5** Save your configuration.

- Step 6** Exit configuration mode.

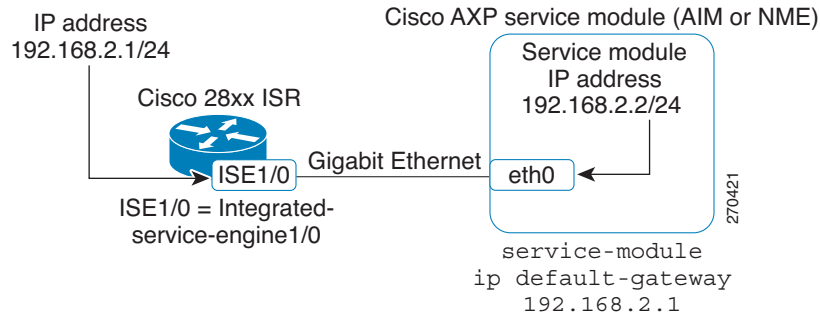
```
2800_w_axp(config-if)# end <enter>
2800_w_axp#
2800_w_axp# copy running-config startup-config <enter>
Destination filename [startup-config]? <enter>
Building configuration...
[OK]
2800_w_axp#
```

- Step 7** Session into the service module.

```
2800_w_axp# service-module Integrated-Service-Engine 1/0 session <enter>
Trying 192.168.2.1, 2066 ... Open <enter>
se-Module>
```

You are now on the Cisco AXP service module. The greater than sign (>) indicates that you are in a mode similar to Cisco IOS privileged EXEC mode.

[Figure 1](#) summarizes the setup and configuration of the Cisco 280x ISR service module.

Figure 1 Cisco 280x ISR and Cisco AXP Service Module Configuration

To familiarize yourself with the command modes, see the [“Packaging Applications on Cisco AXP” section on page 8](#).

Packaging Applications on Cisco AXP

To package your applications on Cisco AXP, perform the steps in the following sections.

- [Preparing the Development Platform, page 8](#)
- [Preparing the Cisco AXP SDK, page 9](#)
- [Preparing an Application with RPMs, page 10](#)
- [Enabling the Linux Shell in a Virtual Instance, page 12](#)
- [Packaging an Application Using the SDK, page 13](#)

Preparing the Development Platform

The Cisco AXP SDK runs most cleanly on Fedora Core 4 Linux. The machines used in examples for this quickstart guide have Fedora Core 4 Linux builds.

Begin your development exercise by setting up the four basic directory structures in the following example.

Preparing the Development Platform: Example

This guide uses the directory structure shown below:

```
opt/
  /axp_#/
    /keys
  /tcptrace
  /output
```

[Table 4](#) defines the directories used in this guide’s examples.

Table 4 Directory Structure in Examples

Directory Name	Definition	Example
sdk-directory	Directory where the developer untars the Cisco AXP SDK.	/opt/axp-sdk.1.0.1/
source-directory	Parent directory where the unpackaged application source files are stored.	/opt/tcptrace/
keys-directory	Directory where development authorization, development certification, and private key are stored.	/opt/axp-sdk.1.0.1/keys/
project-directory	Parent directory where the Cisco AXP package builder tool places execution output (including packages to be installed on the Cisco AXP service module).	/opt/output/

Preparing the Cisco AXP SDK

Preparing Cisco AXP SDK: Example

To prepare the Cisco AXP SDK onto your development machine, perform the following steps.

Step 1 `[root@fedora axp_1.0.1]# tar -xzf sdk.nme.1.0.1.tar.gz <enter>`

This command untars the Cisco AXP SDK file into the /opt/axp-sdk.1.0.1/ directory. The SDK file name is sdk.nme.#.tar, where # is the version number of Cisco AXP SDK software. The version number of the SDK you are using matches must match exactly with the Cisco AXP version number loaded onto the Cisco AXP service module. This Quick Start guide assumes that you are working with Cisco AXP 1.0.1. The Cisco AXP SDK directories are created in the /opt/axp-sdk.1.0.1/ directory.

Step 2 Examine the SDK subdirectory structure.

Step 3 Before you begin running the packaging script, make sure that you have the appropriate certificates loaded onto your system.

This guide assumes that you placed the certificates in the /opt/axp-sdk.1.0.1/keys directory, however, you can choose another directory.

- a. Make a note of where you place the certificates, because the interactive installer will later ask you for the location of the certificates.

```
[root@fedora keys]# ls <enter>
dev_authorization.sig dev_certificate.sig private.key
```

Step 4 Ensure that all three keys in step 3 are present in a readable directory. If you are missing any keys, you cannot continue. Contact your Cisco representative to procure these files.

Step 5 Create an output directory in the /opt directory (or directory of your choice) if you have not already created it. This directory is used by the packaging tool to store the final package files (*.pkg, *.prt1).

Step 6 Enter the **chmod** command to ensure the directory has the correct privileges:

```
[root@fedora opt]# mkdir output <enter>
[root@fedora opt]# chmod 755 output <enter>
```

Step 7 Invoke the package builder tool pkg_build.sh.

You can run the tool from within the `/opt/axp-sdk.1.0.1/tools/` directory or from outside this directory by preceding “`pkg_build.sh`” with the directory path name.

Example 1. (Invoke tool from outside the directory.)

```
[root@fedora /]# /opt/axp-sdk.1.0.1/tools/pkg_build.sh <enter>
```

Example 2. (Invoke tool from within the `/opt/axp-sdk.1.0.1/tools/` directory.)

```
[root@fedora /]# pkg_build.sh <enter>
```

The package builder executes interactively, creating the package files for installation.

- Step 8** Create a manifest file, by copying the output from step 7 (displayed on the screen after the words “Resulting CLI command”), and pasting the output into a file for future use as a script file.

In future, you can use the script file to supply the arguments to the package builder instead of running the tool interactively. This also has the advantage of being able to run the script after you have modified the settings in the file.

Preparing an Application with RPMs

An RPM tool is not supported directly in the Cisco AXP vserver environment by default. To install RPMs as part of your application, you must extract the RPMs in the development environment and then integrate the RPMs with the application before running the packaging utility.

Preparing an Application: Example

- Step 1** Navigate to the tools directory in your Cisco AXP SDK by typing the following:

```
[root@fedora /]# cd /opt/axp-sdk.1.0.1/tools <enter>
[root@fedora tools]# pwd <enter>
/opt/axp_#/tools
[root@fedora tools]# ls <enter>
custom_cli gen_auth.sh pkg_build.sh pkg_bundle.sh pkg_check.sh pkg_info.sh
rpm_extractor.sh
[root@fedora tools]#
```

- Step 2** The `rpm_extractor.sh` tool which exists in the `/opt/axp-sdk.1.0.1/tools` directory, helps you to quickly extract an RPM into a project source root directory, examine any dependencies the RPM requires, and view any pre/post install/uninstall scripts contained in the RPM.

Here is the command line format for `rpm_extractor.sh`:

```
rpm_extractor.sh --proj project-directory [--output] [--scripts] [--deps] RPM-files
```

Required argument:

- **proj** *project-directory*

Location of project directory. The `rpm_extractor.sh` tool creates a sub-directory structure under the project-directory as follows:

```
/project-directory/
  rpm_extractor/
    output/
    scripts/
    deps/
```

- a. The `rpm_extractor.sh` tool places extracted RPM files in the `output/` directory.

- b. The `rpm_extractor.sh` tool places extracted scripts in the `scripts/` directory. The name of the file is the RPM file name suffixed with “.scripts”.
- c. The `rpm_extractor.sh` tool places extracted dependencies in the `deps/` directory. The name of the file is the RPM filename suffixed with “.deps”.



Note If none of the optional arguments are specified, all three of the directories in step 2 are created. These directories are: `output/`, `scripts/`, and `deps/`.

Step 3 Required argument:

RPM-files: List of RPM files, processed simultaneously by the tool, extracted to the same `output/` directory.

Step 4 Optional Arguments:

- **output**: Extract operation to be performed.
- **scripts**: Script extract operation to be performed.
- **deps**: Dependency extract operation to be performed.
- **help**: Print the usage page.

Step 5 Create a directory to catch the output of the `rpm_extractor.sh` tool. This directory can be the same directory in which the packaging tool runs.

Step 6 Invoke the `rpm_extractor.sh` tool from within the `/opt/axp-sdk.1.0.1/tools/` directory or from outside this directory by preceding `rpm_extractor.sh` with the directory path name.

Example 1. (Invoke tool from outside the directory.)

```
[root@fedora tools]# /opt/axp-sdk.1.0.1/tools/rpm_extractor.sh --proj
/opt/rpm_tool_practice/ /opt/tcptrace-6.6.7-1.2.fc5.rf.i386.rpm <enter>
....
<tool-stdout-output>
....
```

Example 2. (Invoke tool from within the `/opt/axp-sdk.1.0.1/tools/` directory.)

```
[root@fedora tools]# ./rpm_extractor.sh --proj /opt/rpm_tool_practice/
opt/tcptrace-6.6.7-1.2.fc5.rf.i386.rpm <enter>
....
<tool stdout output here>
....
```

Step 7 After the tool executes, as shown in the examples in step 6, navigate to the newly created `/opt/rpm_tool_practice/rpm_extractor/` directory to view the output of the `rpm_extractor.sh` tool:

```
[root@fedora tools]# cd /opt/rpm_tool_practice/rpm_extractor/ <enter>
[root@fedora rpm_extractor]# ls <enter>
deps output scripts
```



Note The `rpm_extractor.sh` tool created the three separate directories: `deps`, `output`, and `scripts`.

Step 8 Navigate to the `/opt/rpm_tool_practice/rpm_extractor/deps` directory:

```
[root@fedora rpm_extractor]# cd deps/
```

Step 9 View the contents of the `/opt/rpm_tool_practice/rpm_extractor/deps` directory:

```
[root@fedora deps]# ls
```

```
tcptrace-6.6.7-1.2.fc5.rf.i386.rpm.deps
```

Step 10 The `.deps` file contains all of the dependencies of the RPM file.

To see the dependencies required by the RPM file, type the following:

```
[root@fedora deps]# cat tcptrace-6.6.7-1.2.fc5.rf.i386.rpm.deps /bin/sh <enter>
libc.so.6()
libc.so.6(GLIBC_2.2.5)
libc.so.6(GLIBC_2.3)
libc.so.6(GLIBC_2.3.4)
libc.so.6(GLIBC_2.4)
libm.so.6()
libm.so.6(GLIBC_2.2.5)
libpcap.so.0.9.4()
rpmLib(CompressedFileNames) <= 3.0.4-1
rpmLib(PayloadFilesHavePrefix) <= 4.0-1

[root@fedora deps]#
```

Porting the Application

Use the information on scripts and dependencies as shown in the [“Preparing an Application: Example” section on page 10](#).

To port your application, perform the following steps.

-
- Step 1** Examine the library and RPM dependencies, gather any new required RPM files, and perform the extraction using similar methods to those shown in [“Preparing an Application: Example” section on page 10](#).
 - Step 2** After extracting the RPM files using the `rpm_extractor.sh` tool, copy the output directory to the source directory you are using to package your application by entering the following command:


```
[root@fedora /]# cp /opt/rpm_tool_practice/rpm_extractor/output/*
/opt/source-directory/ <enter>
```
 - Step 3** Examine the pre install, post install, pre uninstall, and post uninstall scripts. Include the appropriate scripts in the project source directory.
 - Step 4** Package the application and install it on the Cisco AXP service module.
-

Enabling the Linux Shell in a Virtual Instance

By default, a Linux shell is not provided in the vserver environment by the CLI.

To configure your application’s virtual instance to have a Linux shell, perform the following steps.

-
- Step 1** Include a postinstall script and pass it to the Cisco AXP packaging tool during execution of the packaging tool.
 - Step 2** Create the postinstall script, such as file `postinstall.sh`, and place the script in a sub-directory under your designated source-directory for your application (the directory is `/opt/tcptrace/` in this guide).

The script is a very simple shell script that tells the Cisco AXP environment that you want to have Linux shell access configured when your application's virtual instance is created.

Step 3 Create the contents of the `postinstall.sh` script file as follows:

```
[root@fedora /]# cat /opt/tcptrace/postscripts/postinstall.sh
#!/bin/sh
ln -s /bin/bash /bin/console
```



Note This script essentially creates a soft link to the console.

Step 4 Change the permissions of non-owner users of the `postinstall` script to read-execute:

```
chmod 755 /opt/tcptrace/postscripts/postinstall.sh
```

Step 5 Navigate to the Linux shell after your application is installed. See “Configuring Console Access” in the [Cisco AXP User Guide](#).

Packaging an Application Using the SDK

Packaging an Application Using the SDK: Example

After your build environment is set up, package the application by performing the following steps.

Step 1 To start the packaging tool in interactive mode, enter the following command from directory `/opt/axp-sdk.1.0.1/tools` :

```
[root@fedora tools]# ./pkg_build.sh
```



Note For details on each parameter of the packaging tool, see the “Packaging Tool” section in the [Cisco AXP Developer Guide](#).

A series of prompts appears.

Step 2 Provide argument values to the packaging tool:

```
Project Directory: /opt/output
Development Certificate File: /opt/axp-sdk.1.0.1/keys/dev_certificate.sig
Development Authorization File: /opt/axp-sdk.1.0.1/lib/keys/dev_authorization.sig
Private Key for Signing Operations: /opt/axp-sdk.1.0.1/keys/private.key
Application Name: tcptrace
Version: 1.0
Description: This is my first Cisco AXP package
Unique Identifier: yourUUID <enter>
Source Directory: /opt/tcptrace
Protect List File: <enter>
Package Name Containing Dependency SSID (blank line when done): <enter>
Disk Limit : 100M
Memory Limit: 100M
CPU Limit : 500
post-install script path: /postscripts/postinstall.sh
```



Note The Source Directory entered above is “/opt/tcptrace”: this is the relative path to the source directory.
The Unique Identifier entered above is “yourUUID”: we recommend you specify this argument with a name of your choice. A file is created with the values of all the arguments specified in this invocation of the packaging tool.

A Cisco AXP package has been created for application *tcptrace*.

You can see the two essential Cisco AXP package files in the project-directory/pkg directory.

These two files are:

- package file: suffixed by .pkg
- installer payload file: suffixed by .prt1

Step 3 If you specified a Project Directory of /opt/output/pkg as shown in Step 2, to view the various artifacts that the packaging tool has created in the /opt/output directory, enter the following command:

```
ls <enter>
```

The following two files are displayed on the screen:

```
tcptrace.1.0.pkg
tcptrace.1.0.prt1
```



Note The Cisco AXP installer, which is resident on the Cisco AXP service module, later expects these two files to be in the same FTP directory.

Step 4 Copy tcptrace.1.0.pkg and tcptrace.1.0.prt1 to the public FTP directory of the machine you are using for your Cisco AXP package FTP repository.

Installing and Uninstalling an Application

This section consists of:

- [Installing an Application, page 14](#)
- [Uninstalling an Application, page 16](#)

Installing an Application

Installing an Application: Example

To install an application, perform the following steps.

Step 1 Make sure that the Cisco Application eXtension Platform (AXP) 1.0.1 is ready on the service module before running the automated installer.

Log into the Cisco 280x ISR if you have not done so already:

```
telnet 192.168.1.39<enter>
2800_w_axp> enable <enter>
```

```
2800_w_axp#service-module Integrated-Service-Engine 1/0 status <enter>
```

The output is similar to the following example:

```
Service Module is Cisco Integrated-Service-Engine1/0
Service Module supports session via TTY line 66
Service Module is in Steady state
Getting status from the Service Module, please wait..
AppRuntimeEnvironment 0.0.6
Online on Bryce
```

Step 2 Ensure that the Cisco AXP service module is in *Steady State* and *Online*.

Step 3 Session into the Cisco AXP service module:

```
2800_w_axp# service-module Integrated-Service-Engine 1/0 session <enter>
se-Module>
```

Step 4 You must ensure that the Cisco AXP service module has sufficient resources to host your application. To view the resources that are currently assigned to all applications, enter the following command:

```
se-Module> show resource limits <enter>
```



Note

Carefully consider the existing resources and future requirements for applications that have not yet been loaded. Plan the capacity requirements at appropriate intervals. See the “Dedicated Application Resources” section in the [Cisco AXP Developer Guide](#).

Step 5 Invoke the installer tool from the CLI.

On the service module, type the following:

```
se-Module> software install add url ftp://192.168.1.1/tcptrace.pkg username username
password password
```

The **software install** command points to the **.pkg* file. The installer uses this package data to download the **.prt1* file, which is the larger of the two files.

After the installer downloads both of these files, it creates a virtual instance on the Cisco AXP service module automatically, and unpacks the contents of the files into the / (root) directory of the virtual instance.



Note

The Cisco AXP service module automatically reboots. This is a normal occurrence.

Step 6 After rebooting, check the application is installed by showing all applications:

```
se-Module> app-service ? <enter>
```

After the application starts you might experience a delay of up to several minutes before this command shows the application as being online.

Step 7 To show the status of all applications, enter the following command:

```
se-Module> show app-service state <enter>
```

Step 8 To enter the virtual instance of the application in this example (“tcptrace”), enter the following command:

```
se-Module> app-service tcptrace <enter>
se-Module>(exec-tcptrace)>
```

You are now in the Cisco AXP application service EXEC mode.

When typing commands you can include the ? character which calls a utility that lists the available commands.

An important command to use during this stage of the application deployment is the **show state** command. The output is the same as that seen from the service module context above, but for convenience, you can also view the state of your application from the Cisco AXP application service EXEC mode.

Step 9 To view the state of your application, enter the following command:

```
se-Module(exec-tcptrace)> show state <enter>
```

Step 10 To view the individual resource allocation (set during the packaging step) for the tcptrace application, enter the following command:

```
se-Module(exec-tcptrace)> show resource limits <enter>
```

This command is for one application and provides little information about overall system resources.



Note

The resource fields that you entered into the interactive packaging tool have been carried forward and are shown by this command.

For more information on installing and upgrading software, see the “Installing and Upgrading Software” section in the [Cisco AXP User Guide](#).

Uninstalling an Application

To uninstall an application, perform the following steps.

Step 1 Use the **software uninstall** command.

```
se-Module> software uninstall <enter>
```

Various application packages are displayed on the screen. The # column identifies each add-on package in the list.

Step 2 To remove your application, type the following, where # is the number of the application as shown in the list.



Note

For the following command, make sure you chose the correct number for your application.

```
Enter Command: r # <enter>
```

Step 3 A confirmation question appears.



Note

You have a second chance to cancel the removal of the application by answering **n** to the question below. After cancelling the uninstall, you can then reenter the original **software uninstall** command from the prompt to check the number of the application you wish to uninstall.

```
Are you sure? [y/n]: y <enter>
```

After you confirm your selection, the system begins removing the application from the Cisco AXP environment.

**Note**

The Cisco AXP service module automatically reboots, which is a normal occurrence.

Step 4

To confirm that the application was removed, wait until the system completes rebooting and the prompt is redisplayed on the screen. Enter the following command to show programs and options that are currently installed:

```
se-Module> show software packages <enter>
```

**Note**

Verify that your application is no longer listed.

Viewing the Application Environment

To view the application environment, use the **show** command in Cisco AXP application service EXEC mode or Cisco AXP EXEC mode.

Use the **show software version** command to view the software versions installed on the service module.

For more information on **show** commands, see the [Cisco AXP 1.0 Command Reference](#) or the “Troubleshooting” section in the [Cisco AXP User Guide](#).

Configuring Network Services

The following sections explain how to configure basic network services for the virtual instance of your application.

- [Binding an Application, page 17](#)
- [Testing IP Connectivity, page 18](#)

Binding an Application

By default, a newly created application virtual instance is bound to interface eth0. Binding ensures that your application has IP connectivity to the Cisco 280x ISR and the rest of the network.

**Note**

If your application is already bound to the eth0 interface, you may skip this section.

The following example describes how to bind the application “tcptrace” to interface eth0 in the Cisco AXP configuration mode.

Step 1 se-Module(exec-tcptrace)> **end** <enter>

Step 2 se-Module> **configure terminal** <enter>

Step 3 To enter the virtual instance where your application resides, type the following command:

```
se-Module(config)> app-service tcptrace <enter>
```

- Step 4** To bind your application to an IP interface allowing IP traffic to your application, enter the following command:

```
se-Module(config-tcptrace)> bind eth0 <enter>
WARNING!!! Reset the hosting environment
WARNING!!! for binding to take effect
```



Note The above warning reminds you to re-enter Cisco AXP EXEC mode after executing the **bind** command.

- Step 5** `se-Module(config-tcptrace)> end <enter>`

- Step 6** `se-Module> app-service tcptrace <enter>`

- Step 7** `se-Module(exec-tcptrace)> reset <enter>`

The system resets. However, you do not see anything displayed on the screen indicating this.

- Step 8** Verify that your virtual instance and application are online by entering the following command:

```
se-Module(exec-tcptrace)> show state <enter>
```

Testing IP Connectivity

Use the **ping** command to test the connectivity between your application's virtual instance to the rest of your network. Each virtual instance has its own Linux shell context and is isolated from other instances and the effects of other applications.

The following example shows how to test IP connectivity to your application using the **ping** command.

- Step 1** To enter into the Linux shell context for your application, make sure you are in the virtual instance of the installed application by entering Cisco AXP application service EXEC mode.

```
se-Module> app-service tcptrace <enter>
se-Module(exec-tcptrace)>
```

- Step 2** Enter the Linux shell for your application (virtual instance shell).

```
se-Module(exec-tcptrace)> connect console <enter>
bash-2.05b#
```

- Step 3** The interface to be pinged in this step is the eth0 interface to which you bound your application using the **bind** command. See step 4 of “[Binding an Application](#)” section on page 17 and [Figure 1](#) on page 8.

Ping the eth0 interface:

```
bash-2.05b# ping 192.168.2.2 <enter>
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=0.063 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=0.061 ms

[1]+  Stopped  ping 192.168.2.2
bash-2.05b#
```



Note Press **Ctrl-C** to stop the ping.

Step 4 Ping the Cisco 280x ISR side of the ISE1/0 interface (router side):

```
bash-2.05b# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=255 time=2.31 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=255 time=0.822 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=255 time=0.799 ms

[2]+ Stopped ping 192.168.2.1
bash-2.05b#
```

Step 5 Ping your FTP server, where your Cisco AXP package files are located:

```
bash-2.05b# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=254 time=3.59 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=254 time=1.68 ms

[1]+ Stopped ping 192.168.1.1
bash-2.05b#
```

**Note**

The ping in this step proves that your application environment has IP network connectivity through the Cisco 280x ISR to your FTP server's interface.

Because your prompt is a standard Linux shell prompt, it appears as if you are logged into a Linux machine with your own root (/) directory.

Step 6 Begin working with your application and see your working vserver environment, network connectivity, and general Linux environment.**Step 7** To exit the Linux shell environment in your virtual instance, type the following:

```
bash-2.05b# exit
se-Module(exec-tcptrace)> exit
se-Module> exit <enter>
2800_w_axp#
```

You are now back to the Cisco 280x ISR prompt.

Where to Go Next

- For details on developing applications, see the [Cisco AXP Developer Guide](#).
- For details on configuring the router and application service module, see the [Cisco AXP User Guide](#).

Time Stamps

If you want to have a consistent set of time stamps required for logs and development authorization between the Cisco 280x ISR and the Cisco AXP service module, you must configure the AXP service module to receive its NTP clock source from the router. See the “Configuring the NTP Server Source” section in the [Cisco AXP User Guide](#).

Appendix A: Cisco 280x ISR and Cisco AXP Service Module Configurations

The running configurations for both the router and service module are described in the following sections.

- [Cisco 280x ISR Running Configuration: Example, page 20](#)
- [Cisco AXP Service Module Running Configuration: Example, page 22](#)

Cisco 280x ISR Running Configuration: Example

In the following example, configuration lines that were referenced and/or configured in this guide are highlighted in **bold**.

```
2800_w_axp# show running-config
Building configuration...

Current configuration : 2540 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname 2800_w_axp
!
boot-start-marker
boot system flash c2800nm-ipbasek9-mz.124-15.T.bin
boot-end-marker
!
logging buffered 51200 warnings
enable secret 5 $1$sRDD$4pbopOZZWg7vWmWgp6XHO.
enable password 7 030752180500
!
no aaa new-model
!
clock timezone EDT -4
ip cef
!
no ip domain lookup
multilink bundle-name authenticated
!
username cisco privilege 15 secret 5 $1$u0Pt$PWzszm0/BUKB962nWPxcy0
username default privilege 15 secret 5 $1$rTGF$iv/hCxsva83qv/5lYetsa1
username sysadmin password 7 050F03092059421D
username ed password 7 0202015D0A130335
!
interface FastEthernet0/0
ip address dhcp
duplex full
speed auto
no mop enabled
!
interface FastEthernet0/1
ip address 192.168.1.39 255.255.255.0
duplex full
speed auto
```

```

no mop enabled
!
interface Integrated-Service-Engine1/0
  ip address 192.168.2.1 255.255.255.0
  service-module ip address 192.168.2.2 255.255.255.0
  service-module ip default-gateway 192.168.2.2
  ntp broadcast destination 192.168.2.1
  no keepalive
no mop enabled
!
ip http server
ip http access-class 23
ip http authentication local
no ip http secure-server
ip http timeout-policy idle 60 life 86400 requests 10000
!
dialer-list 1 protocol ip permit
!
control-plane
!
banner login ^C
-----
Cisco Router and Security Device Manager (SDM) is installed on this device.
This feature requires the one-time use of the username "cisco"
with the password "cisco". The default username and password have a privilege level of 15.

Please change these publicly known initial credentials using SDM or the IOS CLI.
Here are the Cisco IOS commands.

username <myuser> privilege 15 secret 0 <mypassword>
no username cisco

Replace <myuser> and <mypassword> with the username and password you want to use.

For more information about SDM please follow the instructions in the QUICK START
GUIDE for your router or go to http://www.cisco.com/go/sdm
-----
^C
!
line con 0
  login local
line aux 0
line 66
  no activation-character
  no exec
  transport preferred none
  transport input all
  transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
line vty 0 4
  privilege level 15
  password 7 00071A150754
  login local
  transport input telnet
line vty 5 15
  access-class 23 in
  privilege level 15
  password 7 00071A150754
  login local
  transport input telnet
!
scheduler allocate 20000 1000
ntp master
!
end

```

Cisco AXP Service Module Running Configuration: Example

```
se-Module> show running-config
Generating configuration:

clock timezone America/New_York

hostname se-Module

ip domain-name (none)

ntp server 192.168.2.1 prefer

system language preferred "en_US"
software download server url "ftp://127.0.0.1/ftp" credentials hidden
"6u/dKTN/hsEuSAEfw40XlF2eFHnZfyUTSd8ZZNgd+Y9J3x1k2B35j0nfGWTYHfmPSd8ZZNgd+Y9J3x1k2B35j0nfG
WTYHfmPSd8ZZNgd+Y9J3x1k2B35j0nfGWTYHfmP"

log trace local enable

no service password-encryption

interface eth0
 ip address 192.168.2.2 255.255.255.0
 exit

interface eth1
 exit

ip ssh server

app-service tcptrace
 bind interface eth0
 hostname se-Module
 exit

end
se-Module>
```

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:
“This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (ey@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (ey@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

“This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)”.

The word ‘cryptographic’ can be left out if the routines from the library being used are not cryptographic-related.

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: “This product includes software written by Tim Hudson (tjh@cryptsoft.com)”.

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License].

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0807R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008 Cisco Systems, Inc. All rights reserved.

