

To unpack the Cisco AXP SDK in your development machine, perform the following steps.

Contents

- [1 Untar the Cisco AXP SDK file](#)
- [2 Check the keys Subdirectory](#)
- [3 Check the output Subdirectory](#)
- [4 Invoke the Packaging Tool
pkg_build.sh](#)
- [5 Create a Script File \(optional\)](#)
- [6 Obtain the Configuration File](#)

Untar the Cisco AXP SDK file

Example:

In this example, the files within an SDK tar file are uncompressed and extracted to directory /opt/axp-sdk.1.6.1/

The parameters "-xzf" have the following meanings:

x => extract files from the tar file

z => expand compressed files from the tar file

f => indicates that the next argument ("axp-sdk.1.6.1.tar.gz" in this case) is the name of the archive file

```
[root@centos axp_1.6.1]# tar -xzf axp-sdk.1.6.1.tar.gz
```

Unpack the SDK file into the /opt/axp-sdk.1.6.1/ directory. The SDK file name is axp-sdk.x.y.z.tar.gz, where x.y.z is the version number of Cisco AXP SDK software. (For example, 1.6.1) Make sure that the version number of the SDK you are using on your development machine matches the version of Cisco AXP loaded on the Cisco AXP service module. The Cisco AXP SDK directories (for Cisco AXP 1.6.1) are created in the /opt/axp-sdk.1.6.1/ directory.

Check the keys Subdirectory

Before you begin running the packaging script, required certificates need to be loaded into a directory such as: /opt/axp-sdk.1.6.1/keys . During packaging, you will need to enter the location of these certificates.

```
[root@centos keys]# ls
auth_bundle.sig dev_certificate.sig private.key
```

The directory needs to be readable.

Check the output Subdirectory

Create an output directory in the /opt directory (or directory of your choice) if it does not already exist. This directory is used by the packaging tool to store the final packaged files (*.pkg, *.prt1) for the application.

Enter the chmod command to give the directory the correct privileges. Example:

```
[root@centos opt]# mkdir output
[root@centos opt]# chmod 755 output
```

Invoke the Packaging Tool `pkg_build.sh`

You can run the tool either: a) from within the `/opt/axp-sdk.1.6.1/tools/` directory or b) from outside this directory by preceding "`pkg_build.sh`" with the directory path name.

Example 1 (Invoke tool from outside the directory.)

```
[root@centos ~]# /opt/axp-sdk.1.6.1/tools/pkg_build.sh
```

Example 2 (Invoke tool from within the `/opt/axp-sdk.1.6.1/tools/` directory.)

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

In this example, the packaging tool executes interactively, creating the package files for installation.

Create a Script File (optional)

At the end of interactive use of the packaging tool, you can copy the text displayed on the screen after the words "Resulting CLI command" and paste the text into a file for future use as a script file. For example, your "Resulting CLI command" may look like:

```
/opt/axp-sdk.1.6.1/tools/pkg_build.sh --project-dir 'output' --auth-bundle '/opt/axp-sdk.1.6.1/keys/private.key' --private-key '/opt/axp-sdk.1.6.1/keys/private.key' --name 'tcpdump' --version '3.9.4' --description 'tcpdump' --source-dir '/opt/tcpdump' --disk-limit '5000M' --memory-limit '256M' --cpu-limit '7000' --postinstall 'mem,16'
```

In the future, you can use the script file as the basis of a file used to provide arguments to the packaging tool, instead of running the tool interactively.

Obtain the Configuration File

Once the application has been successfully packaged, a configuration file is generated. This will look similar to the following:

Generated files:

```
Output package: /opt/output/pkg/tcpdump.3.9.4.pkg
Output payload: /opt/output/pkg/tcpdump.3.9.4.prt1
Text config file: output/tcpdump.3.9.4.20100614.cfg
```

In the future, you can use this configuration file to generate packages. To do so, use the command

```
<axp sdk location>/tools/pkg_build.sh --file <location of tcpdump.3.9.4.cfg>
```

Return to: [Application Extension Platform 1.x -- Packaging Applications on Cisco AXP](#)

Return to start of Getting Started Guide: [Cisco AXP Getting Started Guide](#)