



# Software and Configurations

---

This chapter describes how to manage the Cisco ASA software and configurations.

- [Upgrade the Software, on page 1](#)
- [Load an Image Using ROMMON, on page 1](#)
- [Load an Image for the ASASM Using ROMMON, on page 3](#)
- [Upgrade the ROMMON Image \(ASA 5506-X, 5508-X, and 5516-X, ISA 3000\), on page 4](#)
- [Recover and Load an Image for the ASA 5506W-X Wireless Access Point, on page 6](#)
- [Downgrade Your Software, on page 6](#)
- [Manage Files, on page 10](#)
- [Set the ASA Image, ASDM, and Startup Configuration, on page 19](#)
- [Back Up and Restore Configurations or Other Files, on page 22](#)
- [Configure Auto Update, on page 38](#)
- [History for Software and Configurations, on page 46](#)

## Upgrade the Software

See the [Cisco ASA Upgrade Guide](#) for full upgrade procedures.

## Load an Image Using ROMMON

To load a software image onto an ASA from the ROMMON mode using TFTP, perform the following steps.

### Procedure

---

- Step 1** Connect to the ASA console port according to the instructions in [Access the Appliance Console](#).
- Step 2** Power off the ASA, then power it on.
- Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
- Step 4** In ROMMON mode, define the interface settings to the ASA, including the IP address, TFTP server address, gateway address, software image file, and port, as follows:

```
rommon #1> interface gigabitethernet0/0
rommon #2> address 10.86.118.4
```

```
rommon #3> server 10.86.118.21
rommon #4> gateway 10.86.118.21
rommon #5> file asa961-smp-k8.bin
```

**Note** Be sure that the connection to the network already exists.

The **interface** command is ignored on the ASA 5506-X, ASA 5508-X, and ASA 5516-X platforms, and you must perform TFTP recovery on these platforms from the Management 1/1 interface.

#### Step 5 Validate your settings:

```
rommon #6> set
ROMMON Variable Settings:
ADDRESS=10.86.118.3
SERVER=10.86.118.21
GATEWAY=10.86.118.21
PORT=GigabitEthernet0/0
VLAN=untagged
IMAGE=asa961-smp-k8.bin
CONFIG=
LINKTIMEOUT=20
PKTTIMEOUT=4
RETRY=20
```

#### Step 6 Ping the TFTP server:

```
rommon #7> ping server
Sending 20, 100-byte ICMP Echoes to server 10.86.118.21, timeout is 4 seconds:

Success rate is 100 percent (20/20)
```

#### Step 7 Save the network settings for future use:

```
rommon #8> sync
Updating NVRAM Parameters...
```

#### Step 8 Load the software image:

```
rommon #9> tftpdnld
ROMMON Variable Settings:
ADDRESS=10.86.118.3
SERVER=10.86.118.21
GATEWAY=10.86.118.21
PORT=GigabitEthernet0/0
VLAN=untagged
IMAGE=asa961-smp-k8.bin
CONFIG=
LINKTIMEOUT=20
PKTTIMEOUT=4
RETRY=20

tftp asa961-smp-k8.bin@10.86.118.21 via 10.86.118.21

Received 14450688 bytes

Launching TFTP Image...
Cisco ASA Security Appliance admin loader (3.0) #0: Mon Mar 5 16:00:07 MST 2016
```

Loading...

After the software image is successfully loaded, the ASA automatically exits ROMMON mode.

- Step 9** Booting the ASA from ROMMON mode does not preserve the system image across reloads; you must still download the image to flash memory. See [Upgrade the Software, on page 1](#).

## Load an Image for the ASASM Using ROMMON

To load a software image to an ASASM from the ROMMON mode using TFTP, perform the following steps.

### Procedure

- Step 1** Connect to the ASA console port according to the instructions in [Access the ASA Services Module Console](#).
- Step 2** Make sure that you reload the ASASM image.
- Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
- Step 4** In ROMMON mode, define the interface settings to the ASASM, including the IP address, TFTP server address, gateway address, software image file, port, and VLAN, as follows:

```
rommon #2> address 10.86.118.4
rommon #3> server 10.86.118.21
rommon #4> gateway 10.86.118.21
rommon #5> file asa961-smp-k8.bin
rommon #5> interface Data0
rommon #6> vlan 1
Data0
Link is UP
MAC Address: 0012.d949.15b8
```

**Note** Be sure that the connection to the network already exists.

- Step 5** Validate your settings:

```
rommon #7> set
ROMMON Variable Settings:
ADDRESS=10.86.118.4
SERVER=10.86.118.21
GATEWAY=10.86.118.21
PORT=Data0
VLAN=1
IMAGE=asa961-smp-k8.bin
CONFIG=
LINKTIMEOUT=20
PKTTIMEOUT=2
RETRY=20
```

- Step 6** Ping the TFTP server:

```
rommon #8> ping server
Sending 20, 100-byte ICMP Echoes to server 10.86.118.21, timeout is 2 seconds:

Success rate is 100 percent (20/20)
```

### Step 7 Load the software image:

```
rommon #9> tftpdnld
Clearing EOBC receive queue ...
cmostime_set = 1
ROMMON Variable Settings:
  ADDRESS=10.86.118.3
  SERVER=10.86.118.21
  GATEWAY=10.86.118.21
  PORT=Data0
  VLAN=1
  IMAGE=asa961-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20

tftp asa961-smp-k8.bin@10.86.118.21 via 10.86.118.21
Starting download. Press ESC to abort.
```

After the software image is successfully loaded, the ASASM automatically exits ROMMON mode.

### Step 8 Booting the module from ROMMON mode does not preserve the system image across reloads; you must still download the image to flash memory. See [Upgrade the Software, on page 1](#).

## Upgrade the ROMMON Image (ASA 5506-X, 5508-X, and 5516-X, ISA 3000)

Follow these steps to upgrade the ROMMON image for the ASA 5506-X series, ASA 5508-X, ASA 5516-X, and ISA 3000. For the ASA models, the ROMMON version on your system must be 1.1.8 or greater. We recommend that you upgrade to the latest version.

You can only upgrade to a new version; you cannot downgrade.



#### Caution

The ASA 5506-X, 5508-X, and 5516-X ROMMON upgrade for 1.1.15 and the ISA 3000 ROMMON upgrade for 1.0.5 takes twice as long as previous ROMMON versions, approximately 15 minutes. **Do not** power cycle the device during the upgrade. If the upgrade is not complete within 30 minutes or it fails, contact Cisco technical support; **do not** power cycle or reset the device.

#### Before you begin

Obtain the new ROMMON image from Cisco.com, and put it on a server to copy to the ASA. The ASA supports FTP, TFTP, SCP, HTTP(S), and SMB servers. Download the image from:

- ASA 5506-X, 5508-X, 5516-X: <https://software.cisco.com/download/home/286283326/type>
- ISA 3000: <https://software.cisco.com/download/home/286288493/type>

## Procedure

**Step 1** Copy the ROMMON image to the ASA flash memory. This procedure shows an FTP copy; enter **copy ?** for the syntax for other server types.

**copy ftp://[username:password@]server\_ip/asa5500-firmware-xxxx.SPA disk0:asa5500-firmware-xxxx.SPA**

**Step 2** To see your current version, enter the **show module** command and look at the Fw Version in the output for Mod 1 in the MAC Address Range table:

```
ciscoasa# show module
[...]
Mod  MAC Address Range                Hw Version  Fw Version  Sw Version
-----
  1  7426.aceb.ccea to 7426.aceb.ccf2  0.3         1.1.5       9.4 (1)
sfr  7426.aceb.cce9 to 7426.aceb.cce9  N/A         N/A
```

**Step 3** Upgrade the ROMMON image:

**upgrade rommon disk0:asa5500-firmware-xxxx.SPA**

### Example:

```
ciscoasa# upgrade rommon disk0:asa5500-firmware-1108.SPA
Verifying file integrity of disk0:/asa5500-firmware-1108.SPA

Computed Hash   SHA2: d824bdeecce1308fc64427367fa559e9
               eefe8f182491652ee4c05e6e751f7a4f
               5cdea28540cf60acde3ab9b65ff55a9f
               4e0cfb84b9e2317a856580576612f4af

Embedded Hash   SHA2: d824bdeecce1308fc64427367fa559e9
               eefe8f182491652ee4c05e6e751f7a4f
               5cdea28540cf60acde3ab9b65ff55a9f
               4e0cfb84b9e2317a856580576612f4af

Digital signature successfully validated
File Name       : disk0:/asa5500-firmware-1108.SPA
Image type      : Release
  Signer Information
    Common Name       : abraxas
    Organization Unit : NCS_Kenton_ASA
    Organization Name : CiscoSystems
    Certificate Serial Number : 553156F4
    Hash Algorithm    : SHA2 512
    Signature Algorithm : 2048-bit RSA
    Key Version       : A
Verification successful.
Proceed with reload? [confirm]
```

**Step 4** Confirm to reload the ASA when you are prompted.

The ASA upgrades the ROMMON image, and then reloads the operating system.

---

## Recover and Load an Image for the ASA 5506W-X Wireless Access Point

To recover and load a software image onto an ASA 5506W-X using TFTP, perform the following steps.

### Procedure

---

**Step 1** Session to the access point (AP) and enter the AP ROMMON (not the ASA ROMMON):

```
ciscoasa# hw-module module wlan recover image
```

**Step 2** Follow the procedure in the [Cisco IOS Software Configuration Guide for Cisco Aironet Access Points](#).

---

## Downgrade Your Software

In many cases, you can downgrade your ASA software and restore a backup configuration from the previous software version. The method of downgrading depends on your ASA platform.

### Guidelines and Limitations for Downgrading

See the following guidelines before downgrading:

- There is no official Zero Downtime Downgrade support for clustering. However, in some cases, Zero Downtime Downgrading will work. See the following known issues for downgrading; note that there may be other issues that require you to reload your cluster units, thus causing downtime.
  - Downgrade to a pre-9.9(1) release with clustering—9.9(1) and later includes an improvement in the backup distribution. If you have 3 or more units in the cluster, you must perform the following steps:
    1. Remove all secondary units from the cluster (so the cluster consists only of the primary unit).
    2. Downgrade 1 secondary unit, and rejoin it to the cluster.
    3. Disable clustering on the primary unit; downgrade it, and rejoin the cluster.
    4. Downgrade the remaining secondary units, and join them back to the cluster, one at a time.
  - Downgrade to a pre-9.9(1) release when you enable cluster site redundancy—You should disable site redundancy if you want to downgrade (or if you want to add a pre-9.9(1) unit to a cluster). Otherwise, you will see side effects, for example, dummy forwarding flows on the unit running the old version.

- Downgrade from 9.8(1) with clustering and crypto-map—There is no Zero Downtime Downgrade support when downgrading from 9.8(1) when you have a crypto-map configured. You should clear the crypto-map configuration before downgrading, and then re-apply the configuration after the downgrade.
- Downgrade from 9.8(1) with clustering unit health check set to .3 to .7 seconds—If you downgrade your ASA software after setting the hold time to .3 - .7 (**health-check holdtime**), this setting will revert to the default of 3 seconds because the new setting is unsupported.
- Downgrade from 9.5(2) or later to 9.5(1) or earlier with clustering (CSCuv82933)—There is no Zero Downtime Downgrade support when downgrading from 9.5(2). You must reload all units at roughly the same time so that a new cluster is formed when the units come back online. If you wait to reload the units sequentially, then they will be unable to form a cluster.
- Downgrade from 9.2(1) or later to 9.1 or earlier with clustering—Zero Downtime Downgrade is not supported.
- Downgrade from 9.10(1) for smart licensing—Due to changes in the smart agent, if you downgrade, you must re-register your device to the Cisco Smart Software Manager. The new smart agent uses an encrypted file, so you need to re-register to use an unencrypted file required by the old smart agent.
- Downgrade to 9.5 and earlier with passwords using PBKDF2 (Password-Based Key Derivation Function 2) hash—Versions before 9.6 do not support PBKDF2 hashing. In 9.6(1), **enable** and **username** passwords longer than 32 characters use PBKDF2 hashing. In 9.7(1), new passwords of all lengths use PBKDF2 hashing (existing passwords continue to use MD5 hashing). If you downgrade, the **enable** password reverts to the default (which is blank). Usernames will not parse correctly, and the **username** commands will be removed. You must re-create your local users.
- Downgrade from Version 9.5(2.200) for the ASAv—The ASAv does not retain the licensing registration state. You need to re-register with the **license smart register idtoken id\_token force** command (for ASDM: see the **Configuration > Device Management > Licensing > Smart Licensing** page, and use the **Force registration** option); obtain the ID token from the Smart Software Manager.
- VPN tunnels are replicated to the standby unit even if the standby unit is running a version of software that does not support the Ciphersuite that the original tunnel negotiated. This scenario occurs when downgrading. In this case, disconnect your VPN connection and reconnect.

## Incompatible Configuration Removed After Downgrading

When you downgrade to an old version, commands that were introduced in later versions will be removed from the configuration. There is no automated way to check the configuration against the target version before you downgrade. You can view when new commands were added in [ASA new features by release](#).

You can view rejected commands *after* you downgrade using the **show startup-config errors** command. If you can perform a downgrade on a lab device, you can preview the effects using this command before you perform the downgrade on a production device.

In some cases, the ASA migrates commands to new forms automatically when you upgrade, so depending on your version, even if you did not manually configure new commands, the downgrade could be affected by configuration migrations. We recommend that you have a backup of your old configuration that you can use when you downgrade. In the case of upgrading to 8.3, a backup is automatically created (<old\_version>\_startup\_cfg.sav). Other migrations do not create back-ups. See the "Version-Specific Guidelines

and Migrations" in the ASA Upgrade guide for more information about automatic command migrations that could affect downgrading.

See also known downgrade issues in [Guidelines and Limitations for Downgrading](#), on page 6.

For example, an ASA running version 9.8(2) includes the following commands:

```
access-list acl1 extended permit sctp 192.0.2.0 255.255.255.0 198.51.100.0 255.255.255.0
username test1 password $sha512$1234$abcdefghijklmnopqrstuvwxy privilege 15
snmp-server user snmpuser1 snmpgroup1 v3 engineID abcdefghijklmnopqrstuvwxyz encrypted auth
md5 12:ab:34 priv aes 128 12:ab:34
```

When you downgrade to 9.0(4), you will see the following errors on startup:

```
access-list acl1 extended permit sctp 192.0.2.0 255.255.255.0 198.51.100.0 255.255.255.0
ERROR: % Invalid input detected at '^' marker.

username test1 password $sha512$1234$abcdefghijklmnopqrstuvwxy pbkdf2 privilege 15
ERROR: % Invalid input detected at '^' marker.

snmp-server user snmpuser1 snmpgroup1 v3 engineID abcdefghijklmnopqrstuvwxyz encrypted auth
md5 12:ab:34 priv aes 128 12:ab:34
ERROR: % Invalid input detected at '^' marker.
```

In this example, support for **sctp** in the **access-list extended** command was added in version 9.5(2), support for **pbkdf2** in the **username** command was added in version 9.6(1), and support for **engineID** in the **snmp-server user** command was added in version 9.5(3).

## Downgrade the Firepower 2100

You can downgrade the ASA software version by restoring the backup configuration to the startup configuration, setting the ASA version to the old version, and then reloading.

### Before you begin

This procedure requires a backup configuration of the ASA before you upgraded, so you can restore the old configuration. If you do not restore the old configuration, you may have incompatible commands representing new or changed features. Any new commands will be rejected when you load the old software version.

### Procedure

#### Step 1

At the ASA CLI, copy the backup ASA configuration to the startup configuration. For failover, perform this step on the active unit. This step replicates the command to the standby unit.

```
copy old_config_url startup-config
```

It's important that you do not save the running configuration to the startup configuration using **write memory**; this command will overwrite your backup configuration.

#### Example:

```
ciscoasa# copy disk0:/9.12.4_cfg.sav startup-config
```



- Step 2** In FXOS, use the Firepower Chassis Manager or FXOS CLI to use the old ASA software version using the upgrade procedure in the [ASA upgrade guide](#) for standalone, failover, or clustering deployments. In this case, specify the old ASA version instead of a new version.
- 

## Downgrade the Firepower 4100/9300

You can downgrade the ASA software version by restoring the backup configuration to the startup configuration, setting the ASA version to the old version, and then reloading.

### Before you begin

- This procedure requires a backup configuration of the ASA before you upgraded, so you can restore the old configuration. If you do not restore the old configuration, you may have incompatible commands representing new or changed features. Any new commands will be rejected when you load the old software version.
- Make sure the old ASA version is compatible with the current FXOS version. If not, downgrade FXOS as the first step before you restore the old ASA configuration. Just make sure the downgraded FXOS is also compatible with the current ASA version (before you downgrade it). If you cannot achieve compatibility, we suggest you do not perform a downgrade.

### Procedure

---

- Step 1** At the ASA CLI, copy the backup ASA configuration to the startup configuration. For failover or clustering, perform this step on the active/control unit. This step replicates the command to the standby/data units.

**copy *old\_config\_url* startup-config**

It's important that you do not save the running configuration to the startup configuration using **write memory**; this command will overwrite your backup configuration.

#### Example:

```
ciscoasa# copy disk0:/9.8.4_cfg.sav startup-config
```

- Step 2** In FXOS, use the Firepower Chassis Manager or FXOS CLI to use the old ASA software version using the upgrade procedure in the [ASA upgrade guide](#) for standalone, failover, or clustering deployments. In this case, specify the old ASA version instead of a new version.

- Step 3** If you are also downgrading FXOS, use the Firepower Chassis Manager or FXOS CLI to set the old FXOS software version to be the current version using the upgrade procedure in the [ASA upgrade guide](#) for standalone, failover, or clustering deployments.
- 

## Downgrade the ASA 5500-X or ISA 3000

The downgrade feature provides a shortcut for completing the following functions on ASA 5500-X and ISA 3000 models:

- Clearing the boot image configuration (**clear configure boot**).
- Setting the boot image to be the old image (**boot system**).
- (Optional) Entering a new activation key (**activation-key**).
- Saving the running configuration to startup (**write memory**). This sets the BOOT environment variable to the old image, so when you reload, the old image is loaded.
- Copying the old configuration backup to the startup configuration (**copy old\_config\_url startup-config**).
- Reloading (**reload**).

### Before you begin

- This procedure requires a backup configuration of the ASA before you upgraded, so you can restore the old configuration.
- Make sure the ASA FirePOWER module version, if installed, is compatible with the old ASA version. You cannot downgrade the FirePOWER module to an earlier major version.

### Procedure

---

ASA 5500-X models only: Downgrade the software and restore the old configuration.

**downgrade** [/noconfirm] *old\_image\_url* *old\_config\_url* [**activation-key** *old\_key*]

#### Example:

```
ciscoasa(config)# downgrade /noconfirm disk0:/asa821-k8.bin disk0:/8_2_1_0_startup_cfg.sav
```

The **/noconfirm** option downgrades without prompting. The *image\_url* is the path to the old image on disk0, disk1, tftp, ftp, or smb. The *old\_config\_url* is the path to the saved, pre-migration configuration. If you need to revert to a pre-8.3 activation key, then you can enter the old activation key.

---

## Manage Files

### View Files in Flash Memory

You can view files in flash memory and see information about files.

#### Procedure

---

**Step 1** View files in flash memory:

```
dir [disk0: | disk1:]
```

**Example:**

```
hostname# dir

Directory of disk0:/
500   -rw-  4958208   22:56:20 Nov 29 2004  cdisk.bin
2513  -rw-   4634     19:32:48 Sep 17 2004  first-backup
2788  -rw-  21601     20:51:46 Nov 23 2004  backup.cfg
2927  -rw-  8670632   20:42:48 Dec 08 2004  asdmfile.bin
```

Enter **disk0:** for the internal flash memory. The **disk1:** keyword represents the external flash memory. The internal flash memory is the default.

**Step 2** View extended information about a specific file:

**show file information** [path:/]filename

**Example:**

```
hostname# show file information cdisk.bin

disk0:/cdisk.bin:
  type is image (XXX) []
  file size is 4976640 bytes version 7.0(1)
```

The file size listed is for example only.

The default path is the root directory of the internal flash memory (disk0:/).

---

## Delete Files from Flash Memory

You can remove files from flash memory that you no longer need.

**Procedure**

---

Delete a file from flash memory:

**delete disk0:** filename

By default, the file is deleted from the current working directory if you do not specify a path. You may use wildcards when deleting files. You are prompted with the filename to delete, and then you must confirm the deletion.

---

## Erase the Flash File System

To erase the flash file system, perform the following steps.

### Procedure

---

- Step 1** Connect to the ASA console port according to the instructions in [Access the ASA Services Module Console](#) or [Access the Appliance Console](#).
- Step 2** Power off the ASA, then power it on.
- Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
- Step 4** Enter the **erase** command, which overwrites all files and erases the file system, including hidden system files:
- ```
rommon #1> erase [disk0: | disk1: | flash:]
```
- 

## Configure File Access

The ASA can use an FTP client, secure copy client, or TFTP client. You can also configure the ASA as a secure copy server so you can use a secure copy client on your computer.

### Configure the FTP Client Mode

The ASA can use FTP to upload or download image files or configuration files to or from an FTP server. In passive FTP, the client initiates both the control connection and the data connection. The server, which is the recipient of the data connection in passive mode, responds with the port number to which it is listening for the specific connection.

### Procedure

---

Set the FTP mode to passive:

```
ftp mode passive
```

**Example:**

```
ciscoasa(config)# ftp mode passive
```

---

### Configure the ASA as a Secure Copy Server

You can enable the secure copy (SCP) server on the ASA. Only clients that are allowed to access the ASA using SSH can establish a secure copy connection.

#### Before you begin

- The server does not have directory support. The lack of directory support limits remote client access to the ASA internal files.
- The server does not support banners or wildcards.
- Enable SSH on the ASA according to [Configure SSH Access](#).

- The ASA license must have the strong encryption (3DES/AES) license to support SSH Version 2 connections.
- Unless otherwise specified, for multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.
- The performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use the **ssh cipher encryption** command; for example, **ssh cipher encryption custom aes128-cbc**

## Procedure

**Step 1** Enable the SCP server:

```
ssh scopy enable
```

**Step 2** (Optional) Manually add or delete servers and their keys from the ASA database:

```
ssh pubkey-chain [no] server ip_address {key-string key_string exit| key-hash {md5 | sha256} fingerprint}
```

### Example:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit
ciscoasa(config-ssh-pubkey-server)# show running-config ssh pubkey-chain
ssh pubkey-chain
  server 10.7.8.9
    key-hash sha256 f1:22:49:47:b6:76:74:b2:db:26:fb:13:65:d8:99:19:
e7:9e:24:46:59:be:13:7f:25:27:70:9b:0e:d2:86:12
```

The ASA stores the SSH host key for each SCP server to which it connects. You can manually manage keys if desired.

For each server, you can specify the **key-string** (public key) or **key-hash** (hashed value) of the SSH host.

The *key\_string* is the Base64 encoded RSA public key of the remote peer. You can obtain the public key value from an open SSH client; that is, from the `.ssh/id_rsa.pub` file. After you submit the Base64 encoded public key, that key is then hashed via SHA-256.

The **key-hash {md5 | sha256} fingerprint** enters the already hashed key (using an MD5 or SHA-256 key); for example, a key that you copied from **show** command output.

**Step 3** (Optional) Enable or disable SSH host key checking. For multiple context mode, enter this command in the admin context.

```
[no] ssh stricthostkeycheck
```

### Example:

```
ciscoasa# ssh stricthostkeycheck
```

```

ciscoasa# copy x scp://cisco@10.86.95.9/x
The authenticity of host '10.86.95.9 (10.86.95.9)' can't be established.
RSA key fingerprint is dc:2e:b3:e4:e1:b7:21:eb:24:e9:37:81:cf:bb:c3:2a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.86.95.9' (RSA) to the list of known hosts.
Source filename [x]?

Address or name of remote host [10.86.95.9]?

Destination username [cisco]?

Destination password []? cisco123

Destination filename [x]?

```

By default, this option is enabled. When this option is enabled, you are prompted to accept or reject the host key if it is not already stored on the ASA. When this option is disabled, the ASA accepts the host key automatically if it was not stored before.

---

### Examples

From a client on the external host, perform an SCP file transfer. For example, in Linux enter the following command:

```

scp -v -pw password [path/]source_filename
username@asa_address:{disk0|disk1}:[path/]dest_filename

```

The **-v** is for verbose, and if **-pw** is not specified, you will be prompted for a password.

The following example adds an already hashed host key for the server at 10.86.94.170:

```

ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.86.94.170
ciscoasa(config-ssh-pubkey-server)# key-hash sha256 65:d9:9d:fe:1a:bc:61:aa:
64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19

```

The following example adds a host string key for the server at 10.7.8.9:

```

ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:
46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit

```

## Configure the ASA TFTP Client Path

TFTP is a simple client/server file transfer protocol, which is described in RFC 783 and RFC 1350 Rev. 2. You can configure the ASA as a TFTP client so that it can copy files to or from a TFTP server. In this way, you can back up and propagate configuration files to multiple ASAs.

This section lets you predefine the path to a TFTP server so you do not need to enter it in commands such as **copy** and **configure net**.

### Procedure

Predefine the TFTP server address and filename for use with **configure net** and **copy** commands:

```
tftp-server interface_name server_ip filename
```

#### Example:

```
ciscoasa(config)# tftp-server inside 10.1.4.7 files/config1.cfg  
ciscoasa(config)# copy tftp: test.cfg
```

```
Address or name of remote host [10.1.4.7]?
```

```
Source filename [files/config1.cfg]?config2.cfg
```

```
Destination filename [test.cfg]?
```

```
Accessing tftp://10.1.4.7/files/config2.cfg;int=outside...
```

You can override the filename when you enter the command; for example, when you use the **copy** command, you can take advantage of the predefined TFTP server address but still enter any filename at the interactive prompts.

For the **copy** command, enter **tftp:** to use the tftp-server value instead of **tftp://url**.

## Copy a File to the ASA

This section describes how to copy the application image, ASDM software, a configuration file, or any other file that needs to be downloaded to internal or external flash memory from a TFTP, FTP, SMB, HTTP, HTTPS, or SCP server.

### Before you begin

- For the IPS SSP software module, before you download the IPS software to disk0, make sure at least 50% of the flash memory is free. When you install IPS, IPS reserves 50% of the internal flash memory for its file system.
- You cannot have two files with the same name but with different letter case in the same directory in flash memory. For example, if you attempt to download the file, Config.cfg, to a location that contains the file, config.cfg, you receive the following error message:

```
%Error opening disk0:/Config.cfg (File exists)
```

- For information about installing the Cisco SSL VPN client, see the *Cisco AnyConnect VPN Client Administrator Guide*. For information about installing Cisco Secure Desktop on the ASA, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administrators*.

- To configure the ASA to use a specific application image or ASDM image if you have more than one installed, or have installed them in external flash memory, see [Set the ASA Image, ASDM, and Startup Configuration, on page 19](#).
- For multiple context mode, you must be in the system execution space.
- (Optional) Specify the interface through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.

## Procedure

---

Copy a file using one of the following server types.

- Copy from a TFTP server:

**copy** [/noconfirm] [interface\_name] **tftp://server[/path]/src\_filename** {**disk0|disk1**} :[/path]/dest\_filename

Example:

```
ciscoasa# copy tftp://10.1.1.67/files/context1.cfg disk0:/context1.cfg
Address or name of remote host [10.1.1.67]?
Source filename [files/context1.cfg]?
Destination filename [context1.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an FTP server:

**copy** [/noconfirm] [interface\_name] **ftp://[user[:password]@]server[/path]/src\_filename** {**disk0|disk1**} :[/path]/dest\_filename

Example:

```
ciscoasa# copy ftp://jcrichton:aeryn@10.1.1.67/files/context1.cfg
disk0:/contexts/context1.cfg
Address or name of remote host [10.1.1.67]?
Source username [jcrichton]?
Source password [aeryn]?
Source filename [files/context1.cfg]?
Destination filename [contexts/context1.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an HTTP(S) server:



```
copy [/noconfirm] [interface_name] http[s]://[user[:password]@]server[:port]/[path]/src_filename
{disk0|disk1}:[path/]dest_filename
```

Example:

```
ciscoasa# copy https://asun:john@10.1.1.67/files/moya.cfg disk0:/contexts/moya.cfg
Address or name of remote host [10.1.1.67]?
Source username [asun]?
Source password [john]?
Source filename [files/moya.cfg]?
Destination filename [contexts/moya.cfg]?
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from an SMB server:

```
copy [/noconfirm] [interface_name] smb://[user[:password]@]server[/path]/src_filename
{disk0|disk1}:[path/]dest_filename
```

Example:

```
ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/test.xml disk0:/test.xml
Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b
!!!!!!!!!!!!!!
11143 bytes copied in 5.710 secs (2228 bytes/sec)
```

- Copy from a SCP server:

The **;int=interface** option bypasses the route lookup and always uses the specified interface to reach the SCP server.

```
copy [/noconfirm] [interface_name]
scp://[user[:password]@]server[/path]/src_filename[;int=interface_name]
{disk0|disk1}:[path/]dest_filename
```

Example:

```
ciscoasa# copy scp://pilot@10.86.94.170/test.cfg disk0:/test.cfg
Address or name of remote host [10.86.94.170]?
Source username [pilot]?
Destination filename [test.cfg]?
The authenticity of host '10.86.94.170 (10.86.94.170)' can't be established.
RSA key fingerprint is
<65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19> (SHA256) .
Are you sure you want to continue connecting (yes/no)? yes
Please use the following commands to add the hash key to the configuration:
ssh pubkey-chain
server 10.86.94.170
```

```

key-hash sha256
65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19

Password: <type in password>
!!!!!!
6006 bytes copied in 8.160 secs (750 bytes/sec)

```

## Copy a File to the Startup or Running Configuration

You can download a text file to the running or startup configuration from a TFTP, FTP, SMB, HTTP(S), or SCP server, or from the flash memory.

### Before you begin

When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

(Optional) Specify the interface through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.

### Procedure

To copy a file to the startup configuration or running configuration, enter one of the following commands for the appropriate download server:

- Copy from a TFTP server:

```
copy [/noconfirm] [interface_name] tftp://server[/path]/src_filename {startup-config | running-config}
```

Example:

```
ciscoasa# copy tftp://10.1.1.67/files/old-running.cfg running-config
```

- Copy from an FTP server:

```
copy [/noconfirm] [interface_name] ftp://[user[:password]@]server[/path]/src_filename {startup-config | running-config}
```

Example:

```
ciscoasa# copy ftp://jcrichton:aeryn@10.1.1.67/files/old-startup.cfg startup-config
```

- Copy from an HTTP(S) server:

```
copy [/noconfirm] [interface_name] http[s]://[user[:password]@]server[:port][/path]/src_filename {startup-config | running-config}
```

Example:

```
ciscoasa# copy https://asun:john@10.1.1.67/files/new-running.cfg running-config
```

- Copy from an SMB server:

```
copy [/noconfirm] [interface_name] smb://[user[:password]@]server[/path]/src_filename  
{startup-config | running-config}
```

Example:

```
ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/new-running.cfg running-config
```

- Copy from a SCP server:

```
copy [/noconfirm] [interface_name]  
scp://[user[:password]@]server[/path]/src_filename[;int=interface_name] {startup-config |  
running-config}
```

Example:

```
ciscoasa# copy scp://pilot:moya@10.86.94.170/new-startup.cfg startup-config
```

The **;int=interface** option bypasses the route lookup and always uses the specified interface to reach the SCP server.

---

## Examples

For example, to copy the configuration from a TFTP server, enter the following command:

```
ciscoasa# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
ciscoasa# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
ciscoasa# copy http://209.165.200.228/configs/startup.cfg startup-config
```

# Set the ASA Image, ASDM, and Startup Configuration

If you have more than one ASA or ASDM image, you should specify the image that you want to boot. If you do not set the image, the default boot image is used, and that image may not be the one intended. For the startup configuration, you can optionally specify a configuration file.

See the following model guidelines:

- Firepower 4100/9300 chassis—ASA upgrades are managed by FXOS; you cannot upgrade the ASA within the ASA operating system, so do not use this procedure for the ASA image. You can upgrade ASA and FXOS separately from each other, and they are listed separately in the FXOS directory listing. The ASA package always includes ASDM.
- Firepower 2100—The ASA, ASDM, and FXOS images are bundled together into a single package. Package updates are managed by FXOS; you cannot upgrade the ASA within the ASA operating system, so do not use this procedure for the ASA image. You *cannot* upgrade ASA and FXOS separately from each other; they are always bundled together.
- ASDM for the Firepower models—ASDM can be upgraded from within the ASA operating system, so you do not need to only use the bundled ASDM image. ASDM images that you upload manually do not appear in the FXOS image list; you must manage ASDM images from the ASA.




---

**Note** When you upgrade the ASA bundle, the ASDM image in the bundle replaces the previous ASDM bundle image on the ASA because they have the same name (**asdm.bin**). But if you manually chose a different ASDM image that you uploaded (for example, **asdm-782.bin**), then you continue to use that image even after a bundle upgrade. To make sure that you are running a compatible version of ASDM, you should either upgrade ASDM before you upgrade the bundle, or you should reconfigure the ASA to use the bundled ASDM image (**asdm.bin**) just before upgrading the ASA bundle.

---

- ASAv—The initial deployment ASAv package puts the ASA image in the read-only boot:/ partition. When you upgrade the ASAv, you specify a different image in flash memory. Note that if you later clear your configuration (**clear configure all**), then the ASAv will revert to loading the original deployment image. The initial deployment ASAv package also includes an ASDM image that it places in flash memory. You can upgrade the ASDM image separately.

See the following default settings:

- ASA image:
  - Physical ASAs—Boots the first application image that it finds in internal flash memory.
  - ASAv—Boots the image in the read-only boot:/ partition that was created when you first deployed.
  - Firepower 4100/9300 chassis—The FXOS system determines which ASA image to boot. You cannot use this procedure to set the ASA image.
  - Firepower 2100—The FXOS system determines which ASA/FXOS package to boot. You cannot use this procedure to set the ASA image.
- ASDM image on all ASAs—Boots the first ASDM image that it finds in internal flash memory, or if one does not exist in this location, then in external flash memory.
- Startup configuration—By default, the ASA boots from a startup configuration that is a hidden file.

## Procedure

---

**Step 1** Set the ASA boot image location:

**boot system** *url*

**Example:**

```
ciscoasa(config)# boot system disk0:/images/asa921.bin
```

The URL can be:

- **{disk0:/ | disk1:/}***[path/]filename*
- **tftp://***[user[:password]@]server[:port]/[path/]filename*

The TFTP option is not supported on all models.

You can enter up to four **boot system** command entries to specify different images to boot from in order; the ASA boots the first image it finds successfully. When you enter the **boot system** command, it adds an entry at the bottom of the list. To reorder the boot entries, you must remove all entries using the **clear configure boot system** command, and re-enter them in the order you desire. Only one **boot system tftp** command can be configured, and it must be the first one configured.

**Note** If the ASA is stuck in a cycle of constant booting, you can reboot the ASA into ROMMON mode. For more information about the ROMMON mode, see [View Debugging Messages](#).

**Step 2** Set the ASDM image to boot:

**asdm image** **{disk0:/ | disk1:/}***[path/]filename*

**Example:**

```
ciscoasa(config)# asdm image disk0:/images/asdm721.bin
```

If you do not specify the image to boot, even if you have only one image installed, then the ASA inserts the **asdm image** command into the running configuration. To avoid problems with Auto Update (if configured), and to avoid the image search at each startup, you should specify the ASDM image that you want to boot in the startup configuration.

**Step 3** (Optional) Set the startup configuration to be a known file instead of the default hidden file:

**boot config** **{disk0:/ | disk1:/}***[path/]filename*

**Example:**

```
ciscoasa(config)# boot config disk0:/configs/startup1.cfg
```

---

# Back Up and Restore Configurations or Other Files

We recommend that you make regular backups of your configuration and other system files to guard against system failure.

## Perform a Complete System Backup or Restoration

These procedures describe how to back up and restore configurations and images to a tar.gz file and transfer it to your local computer.

### Before You Begin Backup or Restore

- You should have at least 300 MB of disk space available at the backup or restore location before you start a backup or restore.
- If you make any configuration changes during or after a backup, those changes will not be included in the backup. If you change a configuration after making the backup, then perform a restore, this configuration change will be overwritten. As a result, the ASA might behave differently.
- You can start only one backup or restore at a time.
- You can only restore a configuration to the same ASA version as when you performed the original backup. You cannot use the restore tool to migrate a configuration from one ASA version to another. If a configuration migration is required, the ASA automatically upgrades the resident startup configuration when it loads the new ASA OS.
- If you use clustering, you can only back up or restore the startup-configuration, running-configuration, and identity certificates. You must create and restore a backup separately for each unit.
- If you use failover, you must create and restore a backup separately for the active and standby units.
- If you set a master passphrase for the ASA, then you need that master passphrase to restore the backup configuration that you create with this procedure. If you do not know the master passphrase for the ASA, see [Configure the Master Passphrase](#) to learn how to reset it before continuing with the backup.
- If you import PKCS12 data (with the **crypto ca trustpoint** command) and the trustpoint uses RSA keys, the imported key pair is assigned the same name as the trustpoint. Because of this limitation, if you specify a different name for the trustpoint and its key pair after you have restored an ASDM configuration, the startup configuration will be the same as the original configuration, but the running configuration will include a different key pair name. This means that if you use different names for the key pair and trustpoint, you cannot restore the original configuration. To work around this issue, make sure that you use the same name for the trustpoint and its key pair.
- You cannot back up using the CLI and restore using ASDM, or vice versa.
- Each backup file includes the following content:
  - Running-configuration
  - Startup-configuration
  - All security images
  - Cisco Secure Desktop and Host Scan images

- Cisco Secure Desktop and Host Scan settings
- AnyConnect (SVC) client images and profiles
- AnyConnect (SVC) customizations and transforms
- Identity certificates (includes RSA key pairs tied to identity certificates; excludes standalone keys)
- VPN pre-shared keys
- SSL VPN configurations
- Application Profile Custom Framework (APCF)
- Bookmarks
- Customizations
- Dynamic Access Policy (DAP)
- Plug-ins
- Pre-fill scripts for connection profiles
- Proxy Auto-config
- Translation table
- Web content
- Version information

## Back Up the System

This procedure describes how to perform a complete system backup.

### Procedure

---

**Step 1** Back up the system:

**backup** [/noconfirm] [context *ctx-name*] [interface *name*] [passphrase *value*] [location *path*]

#### Example:

```
ciscoasa# backup location disk0:/sample-backup
Backup location [disk0:/sample-backup]?
```

If you do not specify the **interface name**, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.

In multiple context mode from the system execution space, enter the **context** keyword to backup the specified context. Each context must be backed up individually; that is, re-enter the **backup** command for each file.

During the backup of VPN certificates and preshared keys, a secret key identified by the **passphrase** keyword is required to encode the certificates. You must provide a passphrase to be used for encoding and decoding the certificates in PKCS12 format. The backup only includes RSA key pairs tied to the certificates and excludes any standalone certificates.

The backup **location** can be a local disk or a remote URL. If you do not provide a location, the following default names are used:

- Single mode—`disk0:hostname.backup.timestamp.tar.gz`
- Multiple mode—`disk0:hostname.context-ctx-name.backup.timestamp.tar.gz`

**Step 2** Follow the prompts:

**Example:**

```
ciscoasa# backup location disk0:/sample-backup
Backup location [disk0:/sample-backup]?

Begin backup..
Backing up [ASA version] ... Done!
Backing up [Running Config] ... Done!
Backing up [Startup Config] ... Done!

Enter a passphrase to encrypt identity certificates. The default is cisco.
You will be required to enter the same passphrase while doing a restore: cisco
Backing up [Identity Certificates] ... Done!

IMPORTANT: This device uses master passphrase encryption. If this backup file
is used to restore to a device with a different master passphrase,
you will need to provide the current master passphrase during restore.
Backing up [VPN Pre-shared keys] ... Done!
Backing up [SSL VPN Configurations: Application Profile Custom Framework] ... Done!
Backing up [SSL VPN Configurations: Bookmarks]... Done!
Backing up [SSL VPN Configurations: Customization] ... Done!
Backing up [SSL VPN Configurations: Dynamic Access Policy] ... Done!
Backing up [SSL VPN Configurations: Plug-in] ... Done!
Backing up [SSL VPN Configurations: Pre-fill scripts for Connection Profile] ... Done!
Backing up [SSL VPN Configurations: Proxy auto-config] ... Done!
Backing up [SSL VPN Configurations: Translation table] ... Done!
Backing up [SSL VPN Configurations: Web Content] ... Done!
Backing up [Anyconnect(SVC) client images and profiles] ... Done!
Backing up [Anyconnect(SVC) customizations and transforms] ... Done!
Backing up [Cisco Secure Desktop and Host Scan images] ... Done!
Backing up [UC-IME tickets] ... Done!
Compressing the backup directory ... Done!
Copying Backup ... Done!
Cleaning up ... Done!
Backup finished!
```

## Restore the Backup

You can specify configurations and images to restore from a zip tar.gz file on your local computer.

### Procedure

**Step 1** Restore the system from the backup file.

**restore** [/noconfirm] [context *ctx-name*] [passphrase *value*] [location *path*]

**Example:**



```
ciscoasa# restore location disk0:/5525-2051.backup.2014-07-09-223$
restore location [disk0:/5525-2051.backup.2014-07-09-223251.tar.gz]?
```

When using the **context** keyword to restore multiple contexts, each backed up context file must be restored individually; that is, re-enter the **restore** command for each file.

**Step 2** Follow the prompts:

**Example:**

```
ciscoasa# restore location disk0:/5525-2051.backup.2014-07-09-223$
restore location [disk0:/5525-2051.backup.2014-07-09-223251.tar.gz]?

Copying Backup file to local disk... Done!
Extracting the backup file ... Done!
Warning: The ASA version of the device is not the same as the backup version,
some configurations might not work after restore!
  Do you want to continue? [confirm] y
Begin restore ...
IMPORTANT: This backup configuration uses master passphrase encryption.
Master passphrase is required to restore running configuration,
startup configuration and VPN pre-shared keys.
Backing up [VPN Pre-shared keys] ... Done!
Backing up [SSL VPN Configurations: Application Profile Custom Framework] ... Done!
Backing up [SSL VPN Configurations: Bookmarks]... Done!
Backing up [SSL VPN Configurations: Customization] ... Done!
Backing up [SSL VPN Configurations: Dynamic Access Policy] ... Done!
Backing up [SSL VPN Configurations: Plug-in] ... Done!
Backing up [SSL VPN Configurations: Pre-fill scripts for Connection Profile] ... Done!
Backing up [SSL VPN Configurations: Proxy auto-config] ... Done!
Backing up [SSL VPN Configurations: Translation table] ... Done!
Backing up [SSL VPN Configurations: Web Content] ... Done!
Backing up [Anyconnect(SVC) client images and profiles] ... Done!
Backing up [Anyconnect(SVC) customizations and transforms] ... Done!
Backing up [Cisco Secure Desktop and Host Scan images] ... Done!
Backing up [UC-IME tickets] ... Done!
Restoring [Running Configuration]
Following messages are as a result of applying the backup running-configuration to
this device, please note them for future reference.

ERROR: Interface description was set by failover and cannot be changed
ERROR: Unable to set this url, it has already been set
Remove the first instance before adding this one
INFO: No change to the stateful interface
Failed to update LU link information
.Range already exists.
WARNING: Advanced settings and commands should only be altered or used
under Cisco supervision.
ERROR: Failed to apply media termination address 198.0.1.228 to interface outside,
the IP is already used as media-termination address on interface outside.
ERROR: Failed to apply media termination address 198.0.0.223 to interface inside,
the IP is already used as media-termination address on interface inside.
WARNING: PAC settings will override http- and https-proxy configurations.
Do not overwrite configuration file if you want to preserve the old http-
and https-proxy configurations.

Cryptochecksum (changed): 98d23c2c ccb31dc3 e51acf88 19f04e28
Done!
Restoring UC-IME ticket ... Done!
Enter the passphrase used while backup to encrypt identity certificates.
The default is cisco. If the passphrase is not correct, certificates will not be restored.
```

```

No passphrase was provided for identity certificates.
Using the default value: cisco. If the passphrase is not correct,
certificates will not be restored.
Restoring Certificates ...
Enter the PKCS12 data in base64 representation....
ERROR: A keypair named Main already exists.
INFO: Import PKCS12 operation completed successfully
. Done!
Cleaning up ... Done!
Restore finished!

```

## Configure Automatic Backup and Restore (ISA 3000)

On the ISA 3000, you can configure automatic backups to a particular location every time you save your configuration using **write memory**.

Automatic restore lets you easily configure new devices with a complete configuration loaded on an SD flash memory card. Automatic restore is enabled in the default factory configuration.

### Configure Automatic Backup (ISA 3000)

On the ISA 3000, you can configure automatic backups to a particular location every time you save your configuration using **write memory**.

#### Before you begin

This feature is only available on the ISA 3000.

#### Procedure

##### Step 1

Set the back-up package parameters:

**backup-package backup** [**interface** *name*] **location** {**diskn**: | *url*} [**passphrase** *string*]

- **interface** *name*—Specifies the interface to reach the backup URL, if you specify off-device storage. If you do not specify the interface name, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.
- **location** {**diskn**: | *url*}—Specifies the storage medium to be used for backing up data. You can specify a URL or local storage. **disk0** is the internal flash drive. **disk1** is an optional USB memory stick on USB 1. **disk2** is an optional USB memory stick on USB 2. And **disk3** is the SD memory card. Note that the default settings for automatic restore use **disk3**.
- **passphrase** *string*—Sets the passphrase to secure the backed-up data. Note that the default settings for automatic restore use "cisco" as the passphrase.

These settings are also used by default with the manual **backup** command. See [Back Up the System, on page 23](#). Note that if you use the manual **backup** command when you have automatic backup or restore enabled, then the system saves a backup file with the specified name, as well as the "auto-backup-asa.tgz" name used by automatic backup and restore.

#### Example:

```
ciscoasa(config)# backup-package backup location disk3: passphrase cisco
```

**Step 2** Enable automatic mode for back-up and restore:

### **backup-package backup auto**

When you save the configuration using **write memory**, the configuration is automatically saved to the backup location as well as to the startup configuration. The backup file has the name "auto-backup-asa.tgz". To disable automatic backups, use the **no** form of the command.

#### **Example:**

```
ciscoasa(config)# backup-package backup auto
```

---

## Configure Automatic Restore (ISA 3000)

Automatic restore mode restores the system configuration on a device without any user intervention. For example, you insert an SD memory card containing a saved backup configuration into a new device and then power the device on. When the device comes up, it checks the SD card to decide if the system configuration needs to be restored. (The restoration is only initiated if the backup file has the "fingerprint" of a different device. The fingerprint of the backup file is updated to match the current device during a backup or restore operation. So if the device has already completed a restore, or if it has created its own backup, then the automatic restore is skipped.) If the fingerprint shows a restoration is required, the device replaces the system configuration (startup-config, running-config, SSL VPN configuration, and so on; see [Back Up the System, on page 23](#) for details about the contents of the backup). When the device finishes booting, it is running the saved configuration.

Automatic restore is enabled in the default factory configuration, so you can easily configure new devices with a complete configuration loaded on an SD memory card without having to perform any pre-configuration of the device.

Because the device needs to decide early in the boot process if the system configuration needs to be restored, it checks ROMMON variables to determine if the device is in automatic restore mode and to obtain the location of the backup configuration. The following ROMMON variables are used:

- **RESTORE\_MODE** = {**auto** | **manual**}

The default is **auto**.

- **RESTORE\_LOCATION** = {**disk0:** | **disk1:** | **disk2:** | **disk3:**}

The default is **disk3:**.

- **RESTORE\_PASSPHRASE** = *key*

The default is **cisco**.

To change the automatic restore settings, complete the following procedure.

#### **Before you begin**

- This feature is only available on the ISA 3000.

- If you use the default restore settings, you need an SD memory card installed (part number SD-IE-1GB=).
- If you need to restore the default configuration to ensure that automatic restore is enabled, use the **configure factory default** command. This command is only available in transparent firewall mode, so if you are in routed firewall mode, use the **firewall transparent** command first.

## Procedure

---

**Step 1** Set the restore package parameters.

**backup-package restore location** {*diskn*: | *url*} [*passphrase string*]

- **location *diskn***:—Specifies the storage medium to be used for restoring data. *disk0* is the internal flash drive. *disk1* is an optional USB memory stick on USB 1. *disk2* is an optional USB memory stick on USB 2. And *disk3* is the SD memory card. The default is *disk3*.
- **passphrase *string***—Sets the passphrase to read the backed-up data. The default is "cisco".

These settings are also used by default with the manual **restore** command. See [Back Up the System, on page 23](#).

### Example:

```
ciscoasa(config)# backup-package restore location disk1: passphrase $upe3rnatural
```

**Step 2** Enable or disable automatic mode for restore.

**[no] backup-package restore auto**

The name of the file that is restored is "auto-backup-asa.tgz".

### Example:

```
ciscoasa(config)# no backup-package restore auto
```

---

## Back up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local flash memory.

### Before you begin

(Optional) Specify the interface through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.

## Procedure

---

Back up the configuring using one of the following server types:

- Copy to a TFTP server:

```
copy [/noconfirm] [interface_name] {startup-config | running-config} tftp://server[/path]/dst_filename
```

Example:

```
ciscoasa# copy running-config tftp://10.1.1.67/files/new-running.cfg
```

- Copy to an FTP server:

```
copy [/noconfirm] [interface_name] {startup-config | running-config}  
ftp://[user[:password]@]server[/path]/dst_filename
```

Example:

```
ciscoasa# copy startup-config ftp://jcrichon:aeryn@10.1.1.67/files/new-startup.cfg
```

- Copy to an SMB server:

```
copy [/noconfirm] [interface_name] {startup-config | running-config}  
smb://[user[:password]@]server[/path]/dst_filename
```

Example:

```
ciscoasa# copy /noconfirm running-config smb://chiana:dargo@10.1.1.67/new-running.cfg
```

- Copy to a SCP server:

```
copy [/noconfirm] [interface_name] {startup-config | running-config}  
scp://[user[:password]@]server[/path]/dst_filename[;int=interface_name]
```

Example:

```
ciscoasa# copy startup-config  
scp://pilot:moya@10.86.94.170/new-startup.cfg
```

The **;int=interface** option bypasses the route lookup and always uses the specified interface to reach the SCP server.

- Copy to the local flash memory:

```
copy [/noconfirm] {startup-config | running-config} {disk0|disk1}:[/path]/dst_filename
```

Example:

```
ciscoasa# copy /noconfirm running-config disk0:/new-running.cfg
```

Be sure that the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

## Back Up a Context Configuration or Other File in Flash Memory

Copy context configurations or other files that are on the local flash memory by entering one of the following commands in the system execution space.

### Before you begin

(Optional) Specify the interface through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the management-only routing table; if there are no matches, it then checks the data routing table.

### Procedure

Back up a context configuration using one of the following server types:

- Copy from flash to a TFTP server:

```
copy [/noconfirm] [interface_name] {disk0|disk1}:[path/]src_filename tftp://server[/path]/dst_filename
```

Example:

```
ciscoasa# copy disk0:/asa-os.bin tftp://10.1.1.67/files/asa-os.bin
```

- Copy from flash to an FTP server:

```
copy [/noconfirm] [interface_name] {disk0|disk1}:[path/]src_filename
```

```
ftp://[user[:password]@]server[/path]/dst_filename
```

Example:

```
ciscoasa# copy disk0:/asa-os.bin ftp://jcrichton:aeryn@10.1.1.67/files/asa-os.bin
```

- Copy from flash to an SMB server:

```
copy [/noconfirm] [interface_name] {disk0|disk1}:[path/]src_filename
```

```
smb://[user[:password]@]server[/path]/dst_filename
```

Example:

```
ciscoasa# copy /noconfirm copy disk0:/asdm.bin  
smb://chiana:dargo@10.1.1.67/asdm.bin
```

- Copy from flash to SCP server:

```
copy [/noconfirm] [interface_name] {disk0|disk1}:[path/]src_filename
```

```
scp://[user[:password]@]server[/path]/dst_filename[;int=interface_name]
```

Example:

```
ciscoasa# copy disk0:/context1.cfg
scp://pilot:moya@10.86.94.170/context1.cfg
```

The **int=interface** option bypasses the route lookup and always uses the specified interface to reach the SCP server.

- Copy from flash to the local flash memory:

```
copy [/noconfirm] {disk0|disk1}:[path/]src_filename {disk0|disk1}:[path/]dst_filename
```

Example:

```
ciscoasa# copy /noconfirm disk1:/file1.cfg disk0:/file1.cfgnew-running.cfg
```

Be sure that the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

---

## Back Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups.

### Procedure

- 
- Step 1** Copy the running configuration to the startup configuration server (connected to the admin context):

```
ciscoasa/contexta# copy running-config startup-config
```

- Step 2** Copy the running configuration to a TFTP server connected to the context network:

```
ciscoasa/contexta# copy running-config tftp:/server[/path]/filename
```

---

## Copy the Configuration from the Terminal Display

### Procedure

- 
- Step 1** Print the configuration to the terminal:

```
more system:running-config
```

**Step 2** Copy the output from this command, and then paste the configuration into a text file.

---

## Back Up Additional Files Using the Export and Import Commands

Additional files essential to your configuration might include the following:

- Files that you import using the **import webvpn** command. Currently, these files include customizations, URL lists, web content, plug-ins, and language translations.
- DAP policies (dap.xml).
- CSD configurations (data.xml).
- Digital keys and certificates.
- Local CA user database and certificate status files.

The CLI lets you back up and restore individual elements of your configuration using the **export** and **import** commands.

To back up these files, for example, those files that you imported with the **import webvpn** command or certificates, perform the following steps.

### Procedure

---

**Step 1** Run the applicable **show** command(s) as follows:

```
ciscoasa # show import webvpn plug-in
ica
rdp
ssh, telnet
vnc
```

**Step 2** Run the **export** command for the file that you want to back up (in this example, the rdp file):

```
ciscoasa # export webvpn plug-in protocol rdp tftp://tftpserver/backupfilename
```

---

## Use a Script to Back Up and Restore Files

You can use a script to back up and restore the configuration files on your ASA, including all extensions that you import via the **import webvpn** CLI, the CSD configuration XML files, and the DAP configuration XML file. For security reasons, we do not recommend that you perform automated backups of digital keys and certificates or the local CA key.

This section provides instructions for doing so and includes a sample script that you can use as is or modify as your environment requires. The sample script is specific to a Linux system. To use it for a Microsoft Windows system, you need to modify it using the logic of the sample.





**Note** You can alternatively use the **backup** and **restore** commands. See [Perform a Complete System Backup or Restoration, on page 22](#) for more information.

## Before You Begin Using Backup and Restore Scripts

To use a script to back up and restore an ASA configuration, first perform the following tasks:

- Install Perl with an Expect module.
- Install an SSH client that can reach the ASA.
- Install a TFTP server to send files from the ASA to the backup site.

Another option is to use a commercially available tool. You can put the logic of this script into such a tool.

## Run the Script

To run a backup-and-restore script, perform the following steps.

### Procedure

- 
- Step 1** Download or cut-and-paste the script file to any location on your system.
  - Step 2** At the command line, enter **Perlscriptname**, where *scriptname* is the name of the script file.
  - Step 3** Press **Enter**.
  - Step 4** The system prompts you for values for each option. Alternatively, you can enter values for the options when you enter the **Perlscriptname** command before you press **Enter**. Either way, the script requires that you enter a value for each option.
  - Step 5** The script starts running, printing out the commands that it issues, which provides you with a record of the CLIs. You can use these CLIs for a later restore, which is particularly useful if you want to restore only one or two files.
- 

## Sample Script

```
#!/usr/bin/perl
#Description: The objective of this script is to show how to back up
configurations/extensions.
# It currently backs up the running configuration, all extensions imported via "import
webvpn" command, the CSD configuration XML file, and the DAP configuration XML file.
#Requirements: Perl with Expect, SSH to the ASA, and a TFTP server.
#Usage: backupasa -option option_value
#      -h: ASA hostname or IP address
#      -u: User name to log in via SSH
#      -w: Password to log in via SSH
#      -e: The Enable password on the security appliance
#      -p: Global configuration mode prompt
#      -s: Host name or IP address of the TFTP server to store the configurations
#      -r: Restore with an argument that specifies the file name. This file is produced
during backup.
#If you don't enter an option, the script will prompt for it prior to backup.
```

```

#
#Make sure that you can SSH to the ASA.

use Expect;
use Getopt::Std;

#global variables
%options=();
$restore = 0; #does backup by default
$restore_file = '';
$aasa = '';
$storage = '';
$user = '';
$password = '';
$enable = '';
$prompt = '';
$date = `date +%F`;
chop($date);
my $exp = new Expect();

getopts("h:u:p:w:e:s:r:", \%options);
do process_options();

do login($exp);
do enable($exp);
if ($restore) {
    do restore($exp, $restore_file);
}
else {
    $restore_file = "$prompt-restore-$date.cli";
    open(OUT, ">$restore_file") or die "Can't open $restore_file\n";
    do running_config($exp);
    do lang_trans($exp);
    do customization($exp);
    do plugin($exp);
    do url_list($exp);
    do webcontent($exp);
    do dap($exp);
    do csd($exp);
    close(OUT);
}
do finish($exp);

sub enable {
    $obj = shift;
    $obj->send("enable\n");
    unless ($obj->expect(15, 'Password:')) {
        print "timed out waiting for Password:\n";
    }
    $obj->send("$enable\n");
    unless ($obj->expect(15, "$prompt#")) {
        print "timed out waiting for $prompt#\n";
    }
}

sub lang_trans {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn translation-table\n");
    $obj->expect(15, "$prompt# ");
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {

```

```

    s/^\s+//;
    s/\s+$//;
    next if /show import/ or /Translation Tables/;
    next unless (/^.\s+.$/);
    ($lang, $transtable) = split(/\s+/, $_);
    $cli = "export webvpn translation-table $transtable language $lang
$storage/$prompt-$date-$transtable-$lang.po";
    $ocli = $cli;
    $ocli =~ s/^\s+export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}
}

sub running_config {
    $obj = shift;
    $obj->clear_accum();
    $cli = "copy /noconfirm running-config $storage/$prompt-$date.cfg";
    print "$cli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub customization {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn customization\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /^\s*$/;
        $cli = "export webvpn customization $_ $storage/$prompt-$date-cust-$_.xml";
        $ocli = $cli;
        $ocli =~ s/^\s+export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub plugin {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn plug-in\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /^\s*$/;
        $cli = "export webvpn plug-in protocol $_ $storage/$prompt-$date-plugin-$_.jar";
        $ocli = $cli;
        $ocli =~ s/^\s+export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
    }
}

```

```

        $obj->expect(15, "$prompt#" );
    }
}

sub url_list {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn url-list\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /\s*$/ or /No bookmarks/;
        $cli="export webvpn url-list $_ $storage/$prompt-$date-urllist-$_.xml";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub dap {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir dap.xml\n");
    $obj->expect(15, "$prompt#" );

    $output = $obj->before();
    return 0 if($output =~ /Error/);

    $cli="copy /noconfirm dap.xml $storage/$prompt-$date-dap.xml";
    $ocli="copy /noconfirm $storage/$prompt-$date-dap.xml disk0:/dap.xml";
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub csd {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir sdesktop\n");
    $obj->expect(15, "$prompt#" );

    $output = $obj->before();
    return 0 if($output =~ /Error/);

    $cli="copy /noconfirm sdesktop/data.xml $storage/$prompt-$date-data.xml";
    $ocli="copy /noconfirm $storage/$prompt-$date-data.xml disk0:/sdesktop/data.xml";
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub webcontent {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn webcontent\n");
    $obj->expect(15, "$prompt#" );
}

```

```

$output = $obj->before();
@items = split(/\n+/, $output);

for (@items) {
    s/^\s+//;
    s/\s+$//;
    next if /show import/ or /No custom/;
    next unless (/^.\s+.$/);
    ($url, $type) = split(/\s+/, $_);
    $turl = $url;
    $turl =~ s/\/\+//;
    $turl =~ s/\+\/-//;
    $cli = "export webvpn webcontent $url $storage/$prompt-$date-$turl";
    $ocli = $cli;
    $ocli =~ s/^export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}
}

sub login {
    $obj = shift;
    $obj->raw_pty(1);
    $obj->log_stdout(0); #turn off console logging.
    $obj->spawn("/usr/bin/ssh $user@$asa") or die "can't spawn ssh\n";
    unless ($obj->expect(15, "password:")) {
        die "timeout waiting for password:\n";
    }

    $obj->send("$password\n");

    unless ($obj->expect(15, "$prompt>")) {
        die "timeout waiting for $prompt>\n";
    }
}

sub finish {
    $obj = shift;
    $obj->hard_close();
    print "\n\n";
}

sub restore {
    $obj = shift;
    my $file = shift;
    my $output;
    open(IN, "$file") or die "can't open $file\n";
    while (<IN>) {
        $obj->send("$_");
        $obj->expect(15, "$prompt#" );
        $output = $obj->before();
        print "$output\n";
    }
    close(IN);
}

sub process_options {
    if (defined($options{s})) {
        $tstr= $options{s};
        $storage = "tftp://$tstr";
    }
}

```

```

else {
    print "Enter TFTP host name or IP address:";
    chop($tstr=<>);
    $storage = "tftp://$tstr";
}
if (defined($options{h})) {
    $asa = $options{h};
}
else {
    print "Enter ASA host name or IP address:";
    chop($asa=<>);
}

if (defined ($options{u})) {
    $user= $options{u};
}
else {
    print "Enter user name:";
    chop($user=<>);
}

if (defined ($options{w})) {
    $password= $options{w};
}
else {
    print "Enter password:";
    chop($password=<>);
}
if (defined ($options{p})) {
    $prompt= $options{p};
}
else {
    print "Enter ASA prompt:";
    chop($prompt=<>);
}
if (defined ($options{e})) {
    $enable = $options{e};
}
else {
    print "Enter enable password:";
    chop($enable=<>);
}

if (defined ($options{r})) {
    $restore = 1;
    $restore_file = $options{r};
}
}

```

## Configure Auto Update

Auto Update is a protocol specification that allows an Auto Update Server to download configurations and software images to many ASAs and can provide basic monitoring of the ASAs from a central location.

### About Auto Update

This section describes how Auto Update is implemented and why you might want to use Auto Update.

## Auto Update Client or Server

The ASA can be configured as either a client or a server. As an Auto Update client, it periodically polls the Auto Update Server for updates to software images and configuration files. As an Auto Update Server, it issues updates for ASAs configured as Auto Update clients.

## Auto Update Benefits

Auto Update is useful in solving many issues facing administrators for ASA management, such as:

- Overcoming dynamic addressing and NAT challenges.
- Committing configuration changes in one action.
- Providing a reliable method for updating software.
- Leveraging well-understood methods for high availability (failover).
- Providing flexibility with an open interface.
- Simplifying security solutions for Service Provider environments.

The Auto Update specification provides the infrastructure necessary for remote management applications to download ASA configurations, software images, and to perform basic monitoring from a centralized location or multiple locations.

The Auto Update specification allows the Auto Update server to either push configuration information and send requests for information to the ASA, or to pull configuration information by having the ASA periodically poll the Auto Update server. The Auto Update server can also send a command to the ASA to send an immediate polling request at any time. Communication between the Auto Update server and the ASA requires a communications path and local CLI configuration on each ASA.

## Auto Update Server Support in Failover Configurations

You can use the Auto Update Server to deploy software images and configuration files to ASAs in an Active/Standby failover configuration. To enable Auto Update on an Active/Standby failover configuration, enter the Auto Update Server configuration on the primary unit in the failover pair.

The following restrictions and behaviors apply to Auto Update Server support in failover configurations:

- Only single mode, Active/Standby configurations are supported.
- When loading a new platform software image, the failover pair stops passing traffic.
- When using LAN-based failover, new configurations must not change the failover link configuration. If they do, communication between the units will fail.
- Only the primary unit will perform the call home to the Auto Update Server. The primary unit must be in the active state to call home. If it is not, the ASA automatically fails over to the primary unit.
- Only the primary unit downloads the software image or configuration file. The software image or configuration is then copied to the secondary unit.
- The interface MAC address and hardware-serial ID is from the primary unit.
- The configuration file stored on the Auto Update Server or HTTP server is for the primary unit only.

## Auto Update Process Overview

The following is an overview of the Auto Update process in failover configurations. This process assumes that failover is enabled and operational. The Auto Update process cannot occur if the units are synchronizing configurations, if the standby unit is in the failed state for any reason other than SSM card failure, or if the failover link is down.

1. Both units exchange the platform and ASDM software checksum and version information.
2. The primary unit contacts the Auto Update Server. If the primary unit is not in the active state, the ASA first fails over to the primary unit and then contacts the Auto Update Server.
3. The Auto Update Server replies with software checksum and URL information.
4. If the primary unit determines that the platform image file needs to be updated for either the active or standby unit, the following occurs:
  - a. The primary unit retrieves the appropriate files from the HTTP server using the URL from the Auto Update Server.
  - b. The primary unit copies the image to the standby unit and then updates the image on itself.
  - c. If both units have new image, the secondary (standby) unit is reloaded first.
    - If hitless upgrade can be performed when secondary unit boots, then the secondary unit becomes the active unit and the primary unit reloads. The primary unit becomes the active unit when it has finished loading.
    - If hitless upgrade cannot be performed when the standby unit boots, then both units reload at the same time.
  - d. If only the secondary (standby) unit has new image, then only the secondary unit reloads. The primary unit waits until the secondary unit finishes reloading.
  - e. If only the primary (active) unit has new image, the secondary unit becomes the active unit, and the primary unit reloads.
  - f. The update process starts again at Step 1.
5. If the ASA determines that the ASDM file needs to be updated for either the primary or secondary unit, the following occurs:
  - a. The primary unit retrieves the ASDM image file from the HTTP server using the URL provided by the Auto Update Server.
  - b. The primary unit copies the ASDM image to the standby unit, if needed.
  - c. The primary unit updates the ASDM image on itself.
  - d. The update process starts again at Step 1.
6. If the primary unit determines that the configuration needs to be updated, the following occurs:
  - a. The primary unit retrieves the configuration file from the using the specified URL.
  - b. The new configuration replaces the old configuration on both units simultaneously.
  - c. The update process begins again at Step 1.



7. If the checksums match for all image and configuration files, no updates are required. The process ends until the next poll time.

## Guidelines for Auto Update

### Context Mode

Auto Update is supported in single context mode only.

### Clustering

No clustering support.

### Models

No support on the following models:

- ASA 5506-X, 5508-X, 5516-X
- Firepower 2100, 4100, and 9300
- ASAv

### Additional Guidelines

- If HTTPS is chosen as the protocol to communicate with the Auto Update server, the ASA uses SSL, which requires the ASA to have a DES or 3DES license.

## Configure Communication with an Auto Update Server

### Procedure

- Step 1** To specify the URL of the Auto Update Server, enter the following command:

```
auto-update server url [source interface] [verify-certificate | no-verification]
```

where *url* has the following syntax:

```
http[s]://[user:password@]server_ip[:port]/pathname
```

The **source interface** keyword and argument specify which interface to use when sending requests to the Auto Update Server. If you specify the same interface specified by the **management-access** command, the Auto Update requests travel over the same IPsec VPN tunnel used for management access.

For HTTPS, the **verify-certificate** keyword (the default) verifies the certificate returned by the Auto Update Server. To disable verification (not recommended), specify the **no-verification** keyword.

- Step 2** (Optional) To identify the device ID to send when communicating with the Auto Update Server, enter the following command:

```
auto-update device-id {hardware-serial | hostname | ipaddress [if-name] | mac-address [if-name] | string text}
```

The identifier used is determined by specifying one of the following parameters:

- The **hardware-serial** argument specifies the ASA serial number.
- The **hostname** argument specifies the ASA hostname.
- The **ipaddress** keyword specifies the IP address of the specified interface. If the interface name is not specified, it uses the IP address of the interface used to communicate with the Auto Update Server.
- The **mac-address** keyword specifies the MAC address of the specified interface. If the interface name is not specified, it uses the MAC address of the interface used to communicate with the Auto Update Server.
- The **string** keyword specifies the specified text identifier, which cannot include white space or the characters ‘, “, , >, & and ?.

**Step 3** (Optional) To specify how often to poll the Auto Update Server for configuration or image updates, enter the following command:

**auto-update poll-period** *poll-period* [*retry-count* [*retry-period*]]

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is zero.

The *retry-period* argument specifies how long to wait (in minutes) between retries. The default is five minutes.

**Step 4** (Optional) To schedule a specific time for the ASA to poll the Auto Update Server, enter the following command:

**auto-update poll-at** *days-of-the-week time* [**randomize** *minutes*] [*retry\_count* [*retry\_period*]]

The *days-of-the-week* argument is any single day or combination of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Other possible values are *daily* (Monday through Sunday), *weekdays* (Monday through Friday), and *weekends* (Saturday and Sunday).

The *time* argument specifies the time in the format HH:MM at which to start the poll. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

The **randomize** *minutes* keyword and argument specify the period to randomize the poll time following the specified start time. The range is from 1 to 1439 minutes.

The *retry\_count* argument specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is zero.

The *retry\_period* argument specifies how long to wait between connection attempts. The default is five minutes. The range is from 1 to 35791 minutes.

**Step 5** (Optional) If the Auto Update Server has not been contacted for a certain period of time, entering the following command causes it to stop passing traffic:

**auto-update timeout** *period*

The *period* argument specifies the timeout period in minutes between 1 and 35791. The default is to never time out (zero minutes). To restore the default, enter the **no** form of this command.

Use the **auto-update timeout** command to be sure that the ASA has the most recent image and configuration. This condition is reported with system log message 201008.

---

### Example

In the following example, an ASA is configured to poll an Auto Update Server with the IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

The ASA is also configured to use the hostname as the device ID and to poll an Auto Update Server every Friday and Saturday night at a random time between 10:00 p.m. and 11:00 p.m. On a failed polling attempt, the ASA will try to reconnect to the Auto Update Server ten times, and will wait three minutes between attempts at reconnecting, as shown in the following example:

```
ciscoasa(config)# auto-update server
https://jcrighton:farscape@209.165.200.224:1742/management source outside verify-certificate
ciscoasa (config)# auto-update device-id hostname
hostname (config)# auto-update poll-at Friday Saturday 22:00 randomize 60 2 10
```

## Configure Client Updates as an Auto Update Server

Entering the **client-update** command enables updates for ASAs configured as Auto Update clients and lets you specify the type of software component (ASDM or boot image), the type or family of ASA, revision numbers to which the update applies, and a URL or IP address from which to obtain the update.

To configure the ASA as an Auto Update Server, perform the following steps.

### Procedure

---

**Step 1** To enable client update, enter the following command:

```
ciscoasa (config)# client-update enable
```

**Step 2** Configure the following parameters for the **client-update** command that you want to apply to the ASAs:

**client-update** {component {asdm | image} | device-id dev\_string | family family\_name | type type} url url-string rev-nums rev-nums}

The **component** {asdm | image} parameter specifies the software component, either ASDM or the boot image of the ASA.

The **device-id** dev\_string parameter specifies a unique string that the Auto Update client uses to identify itself. The maximum length is 63 characters.

The **family** family\_name parameter specifies the family name that the Auto Update client uses to identify itself. It can be asa, pix, or a text string with a maximum length of seven characters.

The **rev-nums** rev-nums parameter specifies the software or firmware images for this client. Enter up to four, in any order, separated by commas.

The **type** *type* parameter specifies the type of clients to notify of a client update. Because this command is also used to update Windows clients, the list of clients includes several Windows operating systems.

The **url** *url-string* parameter specifies the URL for the software/firmware image. This URL must point to a file appropriate for this client. For all Auto Update clients, you must use the protocol “http://” or “https://” as the prefix for the URL.

Configure the parameters for the client update that you want to apply to all ASAs of a particular type. That is, specify the type of ASA and the URL or IP address from which to get the updated image. In addition, you must specify a revision number. If the revision number of the remote ASA matches one of the specified revision numbers, there is no need to update the client, and the update is ignored.

To configure a client update for ASA 5525-Xs, enter the following command:

```
ciscoasa(config)# client-update type asa5525 component asdm url
http://192.168.1.114/aus/asdm710.bin rev-nums 9.10(1)
```

## Monitoring Auto Update

### Monitoring the Auto Update Process

You can use the **debug auto-update client** or **debug fover cmd-exe** commands to display the actions performed during the Auto Update process. The following is sample output from the **debug auto-update client** command.

```
Auto-update client: Sent DeviceDetails to /cgi-bin/dda.pl of server 192.168.0.21
Auto-update client: Processing UpdateInfo from server 192.168.0.21
  Component: asdm, URL: http://192.168.0.21/asdm.bint, checksum:
0x94bced0261cc992ae710faf8d244cf32
  Component: config, URL: http://192.168.0.21/config-rms.xml, checksum:
0x67358553572688a805a155af312f6898
  Component: image, URL: http://192.168.0.21/cdisk73.bin, checksum:
0x6d091b43ce96243e29a62f2330139419
Auto-update client: need to update img, act: yes, stby yes
name
ciscoasa(config)# Auto-update client: update img on stby unit...
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 9001, len = 1024
auto-update: Fover file copy waiting at clock tick 6129280
```

```

fover_parse: Rcvd file copy ack, ret = 0, seq = 4
auto-update: Fover filecopy returns value: 0 at clock tick 6150260, upd time 145980 msec
Auto-update client: update img on active unit...
fover_parse: Rcvd image info from mate
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
Beginning configuration replication: Sending to mate.
auto-update: HA safe reload: reload active waiting with mate state: 50
auto-update: HA safe reload: reload active waiting with mate state: 50

auto-update: HA safe reload: reload active waiting with mate state: 80
      Sauto-update: HA safe reload: reload active unit at clock tick: 6266860
Auto-update client: Succeeded: Image, version: 0x6d091b43ce96243e29a62f2330139419

```

The following syslog message is generated if the Auto Update process fails:

```
%ASA4-612002: Auto Update failed: file version: version reason: reason
```

The *file* is “image”, “asdm”, or “configuration”, depending on which update failed. The *version* is the version number of the update. And the *reason* is the reason that the update failed.

## Monitoring Auto Update Status

See the following command for monitoring Auto Update status:

### show auto-update

The following is sample output from the **show auto-update** command:

```

ciscoasa(config)# show auto-update

Server: https://*****@209.165.200.224:1742/management.cgi?1276
Certificate will be verified
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes
Timeout: none
Device ID: host name [corporate]
Next poll in 4.93 minutes
Last poll: 11:36:46 PST Tue Nov 13 2004
Last PDM update: 23:36:46 PST Tue Nov 12 2004

```

# History for Software and Configurations

| Feature Name                                                   | Platform Releases     | Feature Information                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Secure Copy client                                             | 9.1(5)/9.2(1)         | <p>The ASA now supports the Secure Copy (SCP) client to transfer files to and from a SCP server.</p> <p>We introduced the following commands: <b>ssh pubkey-chain, server (ssh pubkey-chain), key-string, key-hash, ssh stricthostkeycheck.</b></p> <p>We modified the following command: <b>copy scp.</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Configurable SSH encryption and integrity ciphers              | 9.1(7)94(3)95(3)96(1) | <p>Users can select cipher modes when doing SSH encryption management and can configure HMAC and encryption for varying key exchange algorithms. You might want to change the ciphers to be more or less strict, depending on your application. Note that the performance of secure copy depends partly on the encryption cipher used. By default, the ASA negotiates one of the following algorithms in order: 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr. If the first algorithm proposed (3des-cbc) is chosen, then the performance is much slower than a more efficient algorithm such as aes128-cbc. To change the proposed ciphers, use <b>ssh cipher encryption custom aes128-cbc</b>, for example.</p> <p>We introduced the following commands: <b>ssh cipher encryption, ssh cipher integrity</b></p> |
| Auto Update server certificate verification enabled by default | 9.2(1)                | <p>The Auto Update server certificate verification is now enabled by default; for new configurations, you must explicitly disable certificate verification. If you are upgrading from an earlier release, and you did not enable certificate verification, then certificate verification is not enabled, and you see the following warning:</p> <pre>WARNING: The certificate provided by the auto-update servers will not be verified. In order to verify this certificate please use the verify-certificate option.</pre> <p>The configuration will be migrated to explicitly configure no verification.</p> <p><b>auto-update server no-verification</b></p> <p>We modified the following command: <b>auto-update server {verify-certificate   no-verification}.</b></p>                                                                    |

| Feature Name                                   | Platform Releases | Feature Information                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System backup and restore using the CLI        | 9.3(2)            | <p>You can now back up and restore complete system configurations, including images and certificates, using the CLI.</p> <p>We introduced the following commands: <b>backup</b> and <b>restore</b>.</p>                                                                                                                                                                                                                            |
| Recovering and loading a new ASA 5506W-X image | 9.4(1)            | <p>We now support the recovery and loading of a new ASA 5506W-X image.</p> <p>We introduced the following command: <b>hw-module module wlan recover image</b>.</p>                                                                                                                                                                                                                                                                 |
| Automatic Backup and Restore for the ISA 3000  | 9.7(1)            | <p>You can enable auto-backup and/or auto-restore functionality using pre-set parameters in the backup and restore commands. The use cases for these features include initial configuration from external media; device replacement; roll back to an operable state.</p> <p>We introduced the following commands: <b>backup-package location, backup-package auto, show backup-package status, show backup-package summary</b></p> |

