Cisco ACI Configuration Files: Import and Export

Revised: June 22, 2022

New and Changed Information

The following table provides an overview of the significant changes to this guide up to this current release. The table does not provide an exhaustive list of all changes made to the guide or of the new features up to this release.

Table 1: New Features and Changed Behavior in Cisco APIC

Cisco APIC Release	Feature	Description
5.2(3)	Integrity check for exported configuration files that are saved on external servers	There is an integrity check for exported configuration files that are saved on external servers, which ensures that the file's contents are not tampered with. For more information, see Backing up, Restoring, and Rolling Back the Cisco APIC Configuration, on page 13.
4.1(1)	Export Tech Support/Config data with Read-Only Privileges	Configuring an Export Policy Using the GUI, on page 5
4.0(1)	The Cisco Network Assurance Engine (NAE) creates export policies that appear in the GUI	Added a note about the Cisco NAE export policies. • Configuring an Export Policy Using the GUI, on page 5 • Configuring an Export Policy Using the NX-OS Style CLI, on page 7 Configuring an Export Policy Using the REST API, on page 8
2.2(2e)	Applying the **show running config** output to another Cisco APIC	About Import and Export Configurations, on page 22
1.2(1m)	Snapshot and recovery (backing up, restoring, and rolling back) in configuration import/export	Backing up, Restoring, and Rolling Back the Cisco APIC Configuration, on page 13.
1.2(1i)	NX-OS style CLI	Introduced the NX-OS-style CLI.

Overview

This topic provides information on:

- How to use configuration Import and Export to recover configuration states to the last known good state using the Cisco APIC
- How to encrypt secure properties of Cisco APIC configuration files

You can do both scheduled and on-demand backups of user configuration. Recovering configuration states (also known as "roll-back") allows you to go back to a known state that was good before. The option for that is called an Atomic Replace. The configuration import policy (configImportP) supports atomic + replace (importMode=atomic, importType=replace). When set to these values, the imported configuration overwrites the existing configuration, and any existing configuration that is not present in the imported file is deleted. As long as you do periodic configuration backups and exports, or explicitly trigger export with a known good configuration, then you can later restore back to this configuration using the following procedures for the CLI, REST API, and GUI.

For more detailed conceptual information about recovering configuration states using the Cisco APIC, please refer to the *Cisco Application Centric Infrastructure Fundamentals Guide.*

The following section provides conceptual information about encrypting secure properties of configuration files:

Configuration File Encryption

As of release 1.1(2), the secure properties of APIC configuration files can be encrypted by enabling AES-256 encryption. AES encryption is a global configuration option; all secure properties conform to the AES configuration setting. It is not possible to export a subset of the ACI fabric configuration such as a tenant configuration with AES encryption while not encrypting the remainder of the fabric configuration. See the *Cisco Application Centric Infrastructure Fundamentals*, "Secure Properties" chapter for the list of secure properties.

The APIC uses a 16 to 32 character passphrase to generate the AES-256 keys. The APIC GUI displays a hash of the AES passphrase. This hash can be used to see if the same passphrases was used on two ACI fabrics. This hash can be copied to a client computer where it can be compared to the passphrase hash of another ACI fabric to see if they were generated with the same passphrase. The hash cannot be used to reconstruct the original passphrase or the AES-256 keys.

Observe the following guidelines when working with encrypted configuration files:

- Backward compatibility is supported for importing old ACI configurations into ACI fabrics that use the AES encryption configuration option.

Note Reverse compatibility is not supported; configurations exported from ACI fabrics that have enabled AES encryption cannot be imported into older versions of the APIC software.

- Always enable AES encryption when performing fabric backup configuration exports. Doing so will assure that all the secure properties of the configuration will be successfully imported when restoring the fabric.

Note If a fabric backup configuration is exported without AES encryption enabled, none of the secure properties will be included in the export. Since such an unencrypted backup would not include any of the secure properties, it is possible that importing such a file to restore a system could result in the administrator along with all users of the fabric being locked out of the system.

- The AES passphrase that generates the encryption keys cannot be recovered or read by an ACI administrator or any other user. The AES passphrase is not stored. The APIC uses the AES passphrase to generate the AES keys, then discards the passphrase. The AES keys are not exported. The AES keys cannot be recovered since they are not exported and cannot be retrieved via the REST API.

- The same AES-256 passphrase always generates the same AES-256 keys. Configuration export files can be imported into other ACI fabrics that use the same AES passphrase.
- For troubleshooting purposes, export a configuration file that does not contain the encrypted data of the secure properties. Temporarily turning off encryption before performing the configuration export removes the values of all secure properties from the exported configuration. To import such a configuration file that has all secure properties removed, use the import merge mode; do not use the import replace mode. Using the import merge mode will preserve the existing secure properties in the ACI fabric.
- By default, the APIC rejects configuration imports of files that contain fields that cannot be decrypted. Use caution when turning off this setting. Performing a configuration import inappropriately when this default setting is turned off could result in all the passwords of the ACI fabric to be removed upon the import of a configuration file that does not match the AES encryption settings of the fabric.

Note Failure to observe this guideline could result in all users, including fabric administrations, being locked out of the system.

Configuring a Remote Location Using the GUI

This procedure explains how to create a remote location using the APIC GUI.

Procedure

Step 1 On the menu bar, choose **ADMIN > Import/Export**.

Step 2 In the navigation pane, right-click **Remote Locations** and choose **Create Remote Location**.
The **Create Remote Location** dialog appears.

Step 3 Enter the appropriate values in the **Create Remote Location** dialog fields.

Note For an explanation of a field, click the 'i' icon to display the help file.

Step 4 When finished entering values in the **Create Remote Location** dialog fields, click **Submit**.
You have now created a remote location for backing up your data.

Configuring a Remote Location Using the NX-OS Style CLI

In the ACI fabric, you can configure one or more remote destinations for exporting techsupport or configuration files.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	[**no**] **remote path** *remote-path-name* **Example:** `apic1(config)# remote path myFiles`	Enters configuration mode for a remote path.
Step 3	**user** *username* **Example:** `apic1(config-remote)# user admin5`	Sets the user name for logging in to the remote server. You are prompted for a password.
Step 4	**path** {**ftp** | **scp** | **sftp**} *host*[*:port*] [**remote-directory**] **Example:** `apic1(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic`	Sets the path and protocol to the remote server. You are prompted for a password.

Examples

This example shows how to configure a remote path for exporting files.

```
apic1# configure
apic1(config)# remote path myFiles
apic1(config-remote)# user admin5
You must reset the password when modifying the path:
Password:
Retype password:
apic1(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic
You must reset the password when modifying the path:
Password:
Retype password:
```

Configuring a Remote Location Using the REST API

This procedure explains how to create a remote location using the REST API.

```
<fileRemotePath name="local" host="host or ip" protocol="ftp|scp|sftp" remotePath="path to folder"
userName="uname" userPasswd="pwd" />
```

Configuring an Export Policy Using the GUI

This procedure explains how to configure an export policy using the Cisco Application Policy Infrastructure Controller (APIC) GUI. Use the following procedure to trigger a backup of your data.

Note The **Maximum Concurrent Nodes** value that is configured in a scheduler policy determines the number of configuration export policies to act at the time that is specified in the scheduler policy.

For example, if the **Maximum Concurrent Nodes** is set to **1** in a scheduler policy and you have configured two export policies, both utilizing the same scheduler policy, one export policy is successful and other fails. However, if the **Maximum Concurrent Nodes** is set to **2**, both configurations are successful.

When the user is logged in with read-only privileges, Tech Support data can still be exported by right-clicking on the On-Demand Tech Support or Configuration Export polices and choosing **Trigger**.

Procedure

Step 1 On the menu bar, choose **Admin** > **Import/Export**.

Step 2 In the **Navigation** pane, right-click **Export Policies** and choose **Create Configuration Export Policy**.
The **Create Configuration Export Policy** dialog appears.

Step 3 Enter the appropriate values in the **Create Configuration Export Policy** dialog fields.

For an explanation of a field, click the help (**?**) icon to display the help file.

Step 4 After you finish entering values in the **Create Configuration Export Policy** dialog fields, click **Submit**.

You have now created a backup. You can view this under the **Configuration** tab. The backup file will appear in the **Configuration** pane on the right.

Note When deployed and configured to do so, the Cisco Network Assurance Engine (NAE) creates export policies in the Cisco APIC for collecting data at timed intervals. You can identify a Cisco NAE export policy by its name, which is based on the assurance control configuration. If you delete a Cisco NAE export policy in the Cisco APIC, the Cisco NAE export policy will reappear in the Cisco APIC. We recommend that you do not delete the Cisco NAE export policies.

Step 5 In the **Navigation** pane, choose **Export Policies** > **Configuration** > ***policy_name***.

Step 6 In the **Work** pane, choose the **Operational** > **Job Status** tabs.

On this screen, you can view a table with information about the export jobs. If you did not trigger an export job, then the table is empty. The **State** column indicates the status of an export job. The possible values are:

- `success`: The job succeeded.
- `failed`: The job failed.
- `success-with-warnings`: The job succeeded, but there were some issues.

The **Details** column indicates whether the integrity validation succeeded or failed.

If you created a backup, the Cisco APIC creates a file that is shown in the **Operational** view of the backup file that was created. If you want to then import that data, you must create an import policy.

Configuring an Export Policy Using the NX-OS Style CLI

Before you begin

If you want to export snapshots according to a schedule, configure a scheduler before configuring the export policy.

Note The **Maximum Concurrent Nodes** value that is configured in a scheduler policy determines the number of configuration export policies to act at the time that is specified in the scheduler policy.

For example, if the **Maximum Concurrent Nodes** is set to **1** in a scheduler policy and you have configured two export policies, both utilizing the same scheduler policy, one export policy is successful and other fails. However, if the **Maximum Concurrent Nodes** is set to **2**, both configurations are successful.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	[**no**] **snapshot export** *policy-name* **Example:** `apic1(config)# snapshot export myExportPolicy`	Creates a policy for exporting snapshots.
Step 3	**format** {**xml** | **json**} **Example:** `apic1(config-export)# format json`	Specifies the data format for the exported configuration file.
Step 4	(Optional) [**no**] **schedule** *schedule-name* **Example:** `apic1(config-export)# schedule EveryEightHours`	Specifies an existing scheduler for exporting snapshots.
Step 5	(Optional) [**no**] **target** [**infra** | **fabric** | *tenant-name*] **Example:** `apic1(config-export)# target tenantExampleCorp`	Assigns the target of the export, which can be fabric, infra, a specific tenant, or none. If no target is specified, all configuration information is exported. The default is no target.
Step 6	(Optional) [**no**] **remote path** *remote-path-name* **Example:** `apic1(config-export)# remote path myBackupServer`	Specifies the name of a configured remote path to which the file will be sent. If no remote path is specified, the file is exported locally to a folder in the controller. The default is no remote path.
Step 7	**end** **Example:** `apic1(config-export)# end`	Returns to EXEC mode.

	Command or Action	Purpose
Step 8	Required: **trigger snapshot export** *policy-name* **Example:** `apic1# trigger snapshot export myExportPolicy`	Executes the snapshot export task. If the export policy is configured with a scheduler, this step is unnecessary unless you want an immediate export.

Note When deployed, and configured to do so, the Cisco Network Assurance Engine (NAE) also creates export policies in the Cisco APIC for collecting data at timed intervals. You can identify an NAE export policy by its name, which is based on the assurance control configuration. If you delete an NAE export policy in the Cisco APIC, the NAE export policy will reappear in the Cisco APIC. We recommend not deleting the NAE export policies.

Examples

This example shows how to configure the periodic export of a JSON-format snapshot file for a specific tenant configuration.

```
apic1# configure
apic1(config)# snapshot export myExportPolicy
apic1(config-export)# format json
apic1(config-export)# target tenantExampleCorp
apic1(config-export)# schedule EveryEightHours
```

Configuring an Export Policy Using the REST API

This section demonstrates how to configure an export policy using the REST API.

Note The **Maximum Concurrent Nodes** value that is configured in a scheduler policy determines the number of configuration export policies to act at the time that is specified in the scheduler policy.

For example, if the **Maximum Concurrent Nodes** is set to **1** in a scheduler policy and you have configured two export policies, both utilizing the same scheduler policy, one export policy is successful and other fails. However, if the **Maximum Concurrent Nodes** is set to **2**, both configurations are successful.

Procedure

	Command or Action	Purpose

Step 1	To configure an export policy using the REST API:	`POST https://<ip-of-apic>/api/mo/uni/fabric.xml`
		`<fabricInst dn="uni/fabric">`
		`<configExportP name="export" format="xml" adminSt="triggered">`
		`<configRsExportDestination tnFileRemotePathName="backup" />`
		`</configExportP>`
		`<fileRemotePath name="backup" host="10.10.10.1" protocol="scp"`

	Command or Action	Purpose
		`remotePath="/home/user" userName="user" userPasswd="pass" remotePort=22" /> </fabricInst>`
Step 2	To configure an export policy with Read-Only privileges using the REST API: **Example:** `<trigRoProxy name="readAdmin" policyDn=" uni/fabric/configexp-defaultOneTime" adminSt="triggered"/>`	

Note When deployed, and configured to do so, the Cisco Network Assurance Engine (NAE) also creates export policies in the Cisco APIC for collecting data at timed intervals. You can identify an NAE export policy by its name, which is based on the assurance control configuration. If you delete an NAE export policy in the Cisco APIC, the NAE export policy will reappear in the Cisco APIC. We recommend not deleting the NAE export policies.

Configuring an Import Policy Using the GUI

This procedure explains how to configure an Import policy using the APIC GUI. Follow these steps to import your backed up data:

Procedure

Step 1 On the menu bar, choose **ADMIN > Import/Export**.

Step 2 In the navigation pane, right-click **Import Policies** and click **Create Configuration Import Policy**.
The **Create Configuration Import Policy** dialog appears.

Step 3 Enter the appropriate values in the **Create Configuration Import Policy** dialog fields.

Note For an explanation of a field, click the 'i' icon to display the help file. For more detailed information on import types and modes including (**Replace, Merge, Best Effort,** and **Atomic**), refer to the *Cisco Application Centric Infrastructure Fundamentals Guide* .

Step 4 When finished entering values in the **Create Configuration Import Policy** dialog fields, click **Submit**.

Note If you perform a clean reload of the fabric and import a previously-saved configuration, the time zone will change to UTC by default. Reset the time zone to your local time zone after the configuration import for the APIC cluster in these situations.

Configuring an Import Policy Using the NX-OS Style CLI

To configure an import policy using the NX-OS Style CLI, enter the following:

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	[**no**] **snapshot import** *policy-name* **Example:** `apic1(config)# snapshot import myImportPolicy`	Creates a policy for importing snapshots.
Step 3	**file** *filename* **Example:** `apic1(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz`	Specifies the name of the file to be imported.
Step 4	**action** {**merge** | **replace**} **Example:** `apic1(config-import)# action replace`	Specifies whether the imported configuration settings will be merged with the current settings or whether the imported configuration will completely replace the current configuration.
Step 5	[**no**] **mode** {**atomic** | **best-effort**} **Example:** `apic1(config-import)# mode atomic`	Specifies how the import process handles configuration errors when applying the imported settings. The best-effort import mode allows skipping individual configuration errors in the archive, while atomic mode cancels the import upon any configuration error.
Step 6	(Optional) [**no**] **remote path** *remote-path-name* **Example:** `apic1(config-import)# remote path myBackupServer`	Specifies the name of a configured remote path from which the file will be imported. If no remote path is specified, the file is imported locally from a folder in the controller. The default is no remote path.
Step 7	**end** **Example:** `apic1(config-import)# end`	Returns to EXEC mode.
Step 8	Required: **trigger snapshot import** *policy-name* **Example:** `apic1# trigger snapshot import myImportPolicy`	Executes the snapshot import task.

Examples

This example shows how to configure and execute the importing of a snapshot file to replace the current configuration.

```
apic1# show snapshot files
File    : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created : 2015-11-21T01:00:21.167+00:00
Root    :
Size    : 22926
```

```
apic1# configure
apic1(config)# snapshot import myImportPolicy
apic1(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-import)# action replace
apic1(config-import)# mode atomic
apic1(config-import)# end
apic1# trigger snapshot import myImportPolicy
```

Configuring an Import Policy Using the REST API

To configure an import policy using the REST API:

```
POST
https://<ip-of-apic>/api/mo/uni/fabric.xml
<fabricInst dn="uni/fabric">
<configImportP name="imp" fileName="aa.tar.gz" adminSt="triggered" importType="replace"
importMode="best-effort">
<configRsImportSource tnFileRemotePathName="backup" />
</configImportP>
<fileRemotePath name="backup" host="10.10.10.1" protocol="scp"
remotePath="/home/user" userName="user" userPasswd="pass" />
</fabricInst>
```

Encrypting Configuration Files Using the GUI

AES-256 encryption is a global configuration option. When enabled, all secure properties conform to the AES configuration setting. A portion of the ACI fabric configuration can be exported using configuration export with a specific targetDn. However, it is not possible to use REST API to export just a portion of the ACI fabric such as a tenant configuration with secure properties and AES encryption. The secure properties do not get included during REST API requests.

This section explains how to enable AES-256 encryption.

Procedure

Step 1 On the menu bar, choose **ADMIN > AAA**.

Step 2 In the navigation pane, click **AES Encryption Passphrase and Keys for Config Export (and Import)**.
The **Global AES Encryption Settings for all Configurations Import and Export** window appears in the right pane.

Step 3 Create a passphrase, which can be between 16 and 32 characters long. There are no restrictions on the type of characters used.

Step 4 Click **SUBMIT**.

Note Once you have created and posted the passphrase, the keys are then generated in the back-end and the passphrase is not recoverable. Therefore, your passphrase is not visible to anyone because the key is automatically generated then deleted. Your backup only works if you know the passphrase (no one else can open it).

The **Key Configured** field now shows **yes**. You now see an encrypted hash (which is not the actual passphrase, but just a hash of it) in the **Encrpyted Passphrase** field.

Step 5 After setting and confirming your passphrase, check the check box next to **Enable Encryption** to turn the AES encryption feature on (checked).

The **Global AES Encryption Settings** field in your export and import policies will now be enabled by default.

Note

- Be sure that the **Fail Import if secure fields cannot be decrypted** check box is checked (which is the default selection) in your import and export policies. We highly recommend that you do not uncheck this box when you import configurations. If you uncheck this box, the system attempts to import all the fields However, any fields that it cannot encrypt are blank/missing. As a result, you could lock yourself out of the system because the admin passwords could go blank/missing (if you lock yourself out of the system, refer to *Cisco APIC Troubleshooting Guide*). Unchecking the box launches a warning message. If the box is checked, there are security checks that prevent lockouts and the configuration does not import.
- When the **Enable Encryption** check box is unchecked (off), encryption is disabled and all exported configurations (exports) are missing the secure fields (such as passwords and certificates). When this box is checked (on), encryption is enabled and all exports show the secure fields.
- After enabling encryption, you cannot configure a passphrase when creating a new import or export policy. The passphrase you previously set is now global across all configurations in this box and across all tenants. If you export a configuration from this tab (you have configured a passphrase and enabled encryption) you get a complete backup file. If encryption is not enabled, you get a backup file with the secure properties removed. These backup files are useful when exporting to TAC support engineers, for example, because all the secure fields are missing. This is true for any secure properties in the configuration. There is also a clear option that clears the encryption key.

Note the list of the configuration import behaviors and associated results in the following table:

Configuration Import Behavior Scenario	**Result**
Old configuration from previous release	Import of configurations from old releases is fully supported and successfully imports all secure fields stored in old configurations.
Configuration import when AES encryption is not configured	If the import is for a configuration without secure fields, it is successful with the behavior previously described. If the imported configuration has secure fields, it is rejected.
Configuration import when AES passphrases do not match	If the import is for a configuration without secure fields, it is successful with the behavior previously described. If the imported configuration has secure fields, it is rejected.
Configuration import when AES passphrases match	Import is successful
Configuration import when AES passphrases do not match for copy/pasted fields	This specific case occurs when you have copied and pasted secure fields from other configurations that were exported with a different passphrase. During the first pass parsing of the imported backup file, if any property fails to decrypt correctly, the import fails without importing any shards. Therefore, if a shard fails to decrypt all properties, all shards are rejected.

Encrypting Configuration Files Using the NX-OS Style CLI

To encrypt a configuration file using the NX-OS Style CLI:

```
apic1# configure
apic1(config)# crypto aes
 <CR>
apic1(config)# crypto aes
apic1(config-aes)#
 clear-encryption-key  Clears AES encryption key
encryption            Enable AES Encryption
no                    Negate a command or set its defaults
passphrase            Configure passphrase for AES encryption

bash                  bash shell for unix commands
end                   Exit to the exec mode
exit                  Exit from current mode
fabric                show fabric related information
show                  Show running system information
where                 show the current mode
apic1(config-aes)# encryption
 <CR>
apic1(config-aes)# encryption
apic1(config-aes)#
 clear-encryption-key  Clears AES encryption key
encryption            Enable AES Encryption
no                    Negate a command or set its defaults
passphrase            Configure passphrase for AES encryption

bash                  bash shell for unix commands
end                   Exit to the exec mode
exit                  Exit from current mode
fabric                show fabric related information
show                  Show running system information
where                 show the current mode
apic1(config-aes)# passphrase
 WORD  Passphrase for AES encryption (Range of chars: 16-32) in quotes
apic1(config-aes)# passphrase "abcdefghijklmnopqrstuvwxyz"
apic1(config-aes)#
```

Encrypting Configuration Files Using the REST API

Procedure

To encrypt a configuration file using the REST API, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/mo/uni/fabric.xml
<pkiExportEncryptionKey passphrase="abcdefghijklmnopqrstuvwxyz" strongEncryptionEnabled="true"/>
```

Backing up, Restoring, and Rolling Back the Cisco APIC Configuration

This section describes the set of features for backing up (creating snapshots), restoring, and rolling back a Cisco Application Policy Infrastructure Controller's (APIC's) configuration.

Beginning with the 5.2(3) release, when you export a configuration file to an external server, the Cisco APIC calculates the MD5 checksum for the file contents and stores it in a MD5 file. This MD5 file gets exported along with the configuration file. When importing the configuration file, the Cisco APIC validates the file's integrity by comparing its current MD5 checksum with the value

stored on the MD5 file, and the Cisco APIC informs you whether the integrity validation succeeds or fails. By default, this feature is enabled.

Backing Up, Restoring, and Rolling Back Configuration Files Workflow

This section describes the workflow of the features for backing up, restoring, and rolling back configuration files. All of the features described in this document follow the same workflow pattern. Once the corresponding policy is configured, **admintSt** must be set to **triggered** in order to trigger the job.

Once triggered, an object of type **configJob** (representing that run) is created under a container object of type **configJobCont**. (The naming property value is set to the policy DN.) The container's **lastJobName** field can be used to determine the last job that was triggered for that policy.

Note Up to five **configJob** objects are kept under a single job container at a time, with each new job triggered. The oldest job is removed to ensure this.

The **configJob** object contains the following information:

- Execution time
- Name of the file being processed/generated
- Status, as follows:
  - Pending
  - Running
  - Failed
  - Fail-no-data
  - Success
  - Success-with-warnings
- Details string (failure messages and warnings)
- Progress percentage = 100 * lastStepIndex/totalStepCount
- Field lastStepDescr indicating what was being done last

Configuration Export to Controller

The configuration export extracts user-configurable managed object (MO) trees from all thirty-two shards in the cluster, writes them into separate files, then compresses them into a tar gzip. The configuration export then uploads the tar gzip to a pre-configured remote location (configured using **configRsRemotePath** pointing to a **fileRemotePath** object) or stores it as a **snapshot** on the controller(s).

Note See the Snapshots section for more details.

The **configExportP** policy is configured as follows:

- **name**: Policy name.

- **format**: Format in which the data is stored inside the exported archive (xml or json).
- **targetDn**: The domain name (DN) of the specific object you want to export (empty means everything).
- **snapshot**: When set to `True`, the file is stored on the controller, no remote location configuration is needed.
- **includeSecureFields**: Set to true by default, indicates whether the encrypted fields (passwords, etc.) should be included in the export archive.

Note The **configSnapshot** object is created holding the information about this snapshot (see the Snapshots section).

Scheduling Exports

An export policy can be linked with a scheduler, which triggers the export automatically based on a pre-configured schedule. This is done via the **configRsExportScheduler** relation from the policy to a **trigSchedP** object (see the following Sample Configuration section).

Note A scheduler is optional. A policy can be triggered at any time by setting the **adminSt** to **triggered**.

Troubleshooting

If you get an error message indicating that the generated archive could not be uploaded to the remote location, refer to the Connectivity Issues section.

Sample Configuration Using the NX-OS Style CLI

The following is a sample configuration using the NX-OS Style CLI:

```
apic1(config)# snapshot
download  Configuration snapshot download setup mode
export    Configuration export setup mode
import    Configuration import setup mode
rollback  Configuration rollback setup mode
upload    Configuration snapshot upload setup mode
apic1(config)# snapshot export policy-name
apic1(config-export)#
format    Snapshot format: xml or json
no        Negate a command or set its defaults
remote    Set the remote path configuration will get exported to
schedule  Schedule snapshot export
target    Snapshot target

bash      bash shell for unix commands
end       Exit to the exec mode
exit      Exit from current mode
fabric    show fabric related information
show      Show running system information
where     show the current mode
apic1(config-export)# format xml
apic1(config-export)# no remote path         [If no remote path is specified, the file is exported locally to
 a folder in the controller]
apic1(config-export)# target                 [Assigns the target of the export, which can be fabric, infra, a
 specific tenant, or none. If no target is specified, all configuration information is exported.]
WORD  infra, fabric or tenant-x
```

```
apic1(config-export)#
apic1# trigger snapshot export policy-name   [Executes the snapshot export task]
apic1# ls /data2                            [If no remote path is specified, the configuration export file is
 saved locally to the controller under the folder data2]
ce_Dailybackup.tgz
```

Sample Configuration Using the GUI

The following is a sample configuration using the GUI:

1. On the menu bar, click the **Admin** tab.
2. Choose **IMPORT/EXPORT**.
3. Under **Export Policies**, choose **Configuration**.
4. Under Configuration, click the configuration that you would like to roll back to. For example, you can click **defaultOneTime**, which is the default.
5. Next to **Format**, choose a button for either JSON or XML format.
6. Next to **Start Now**, choose a button for either **No** or **Yes** to indicate whether you want to trigger now or trigger based on a schedule. The easiest method is to choose to trigger immediately.
7. For the **Target DN** field, enter the name of the tenant configuration you are exporting.
8. If you want to store the configuration on the controller itself, check the **Snapshot** option. If you want to configure a remote location, uncheck this option.
9. For the **Scheduler** field, you have the option to create a scheduler instructing when and how often to export the configuration.
10. For the **Encryption** field, you have the option to enable or disable the encryption of your configuration file.
11. When you have finished your configuration, click **Start Now**.
12. Click **Submit** to trigger your configuration export.

Sample Configuration Using REST API

The following is a sample configuration using the REST API:

```
<configExportP name="policy-name" format="xml" targetDn="/some/dn or empty which means everything"
snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
<configRsExportScheduler tnTrigSchedPName="some scheduler name" />
</configExportP>
```

Note When providing a remote location, if you set the snapshot to `True`, the backup ignores the remote path and stores the file on the controller.

Configuration Import to Controller

Configuration import downloads, extracts, parses, analyzes and applies the specified, previously exported archive one shard at a time in the following order: infra, fabric, tn-common, then everything else. The fileRemotePath configuration is performed the same way as for export (via configRsRemotePath). Importing snapshots is also supported.

The **configImportP** policy is configured as follows:

- **name** - policy name
- **fileName** - name of the archive file (not the path file) to be imported
- **importMode**
  - Best-effort mode: each MO is applied individually, and errors only cause the invalid MOs to be skipped.

    Note If the object is not present on the controller, none of the children of the object get configured. Best-effort mode attempts to configure the children of the object.

  - Atomic mode: configuration is applied by whole shards. A single error causes whole shard to be rolled back to its original state.
- **importType**
  - replace - Current system configuration is replaced with the contents or the archive being imported (only atomic mode is supported)
  - merge - Nothing is deleted, archive content is applied on top the existing system configuration.
- **snapshot** - when true, the file is taken from the controller and no remote location configuration is needed.
- **failOnDecryptErrors** - (true by default) the file fails to import if the archive was encrypted with a different key than the one that is currently set up in the system.

Troubleshooting

The following scenarios may need troubleshooting:

- If the generated archive could not be downloaded from the remote location, refer to the Connectivity Issues section.
- If the import succeeded with warnings, check the details.
- If a file could not be parsed, refer to the following scenarios:
  - If the file is not a valid XML or JSON file, check whether or not the files from the exported archive were manually modified.
  - If an object property has an unknown property or property value, it may be because:
    - The property was removed or an unknown property value was manually entered
    - The model type range was modified (non-backward compatible model change)
    - The naming property list was modified
- If an MO could not be configured, note the following:
  - Best-effort mode logs the error and skips the MO
  - Atomic mode logs the error and skips the shard

Sample Configuration Using the NX-OS Style CLI

The following is a sample configuration using the NX-OS Style CLI:

```
apic1# configure
apic1(config)# snapshot
 download  Configuration snapshot download setup mode
export    Configuration export setup mode
import    Configuration import setup mode
rollback  Configuration rollback setup mode
upload    Configuration snapshot upload setup mode
apic1(config)# snapshot import
 WORD       Import configuration name
default
rest-user
apic1(config)# snapshot import policy-name
apic1(config-import)#
 action  Snapshot import action merge|replace
file    Snapshot file name
mode    Snapshot import mode atomic|best-effort
no      Negate a command or set its defaults
remote  Set the remote path configuration will get imported from

bash    bash shell for unix commands
end     Exit to the exec mode
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-import)# file < from "show snapshot files" >
apic1(config-import)# no remote path
apic1(config-import)#
apic1# trigger snapshot import policy-name       [Executes the snapshot import task]
```

Sample Configuration Using the GUI

The following is a sample configuration using the GUI:

1. On the menu bar, click the **ADMIN** tab.
2. Select **IMPORT/EXPORT**.
3. Under **Import Policies**, select **Configuration**.
4. Under **Configuration**, select **Create Configuration Import Policy**. The **CREATE CONFIGURATION IMPORT POLICY** window appears.
5. In the **Name** field, the file name must match whatever was backed up and will have a very specific format. The file name is known to whoever did the backup.
6. The next two options relate to recovering configuration states (also known as "roll-back"). The options are **Input Type** and **Input Mode**. When you recover a configuration state, you want to roll back to a known state that was good before. The option for that is an **Atomic Replace**.
7. If you want to store the configuration on the controller itself, check the **Snapshot** option. If you want to configure a remote location, uncheck this option.
8. In the **Import Source** field, specify the same remote location that you already created.
9. For the **Encryption** field, you have the option to enable or disable the encryption

   of your configuration file.
10. Click **SUBMIT** to trigger your configuration import.

Sample Configuration Using the REST API

The following shows a sample configuration using the REST API:

```
<configImportP name="policy-name" fileName="someexportfile.tgz" importMode="atomic" importType="replace"
snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configImportP>
```

Snapshots

Snapshots are configuration backup archives, stored (and replicated) in a controller managed folder. To create one, an export can be performed with the **snapshot** property set to true. In this case, no remote path configuration is needed. An object of **configSnapshot** type is created to expose the snapshot to the user.

You can create recurring snapshots, which are saved to **Admin** > **Import/Export** > **Export Policies** > **Configuration** > **defaultAuto**.

configSnapshot objects provide the following:

- file name
- file size
- creation date
- root DN indicating what the snapshot is of (fabric, infra, specific tenant, and so on)
- ability to remove a snapshot (by setting the retire field to true)

To import a snapshot, first create an import policy. Navigate to **Admin** > **Import/Export** and click **Import Policies**. Right click and choose **Create Configuration Import Policy** to set the import policy attributes.

Snapshot Manager Policy

The **configSnapshotManagerP** policy allows you to create snapshots from remotely stored export archives. You can attach a remote path to the policy, provide the file name (same as with configImportP), set the mode to download, and trigger. The manager downloads the file, analyzes it to make sure the archive is valid, stores it on the controller, and creates the corresponding configSnapshot object.

You can also create a recurring snapshot.

Note When enabled, recurring snapshots are saved to **Admin** > **Import/Export** > **Export Policies** > **Configuration** > **defaultAuto**.

The snapshot manager also allows you to upload a snapshot archive to a remote location. In this case, the mode must be set to upload.

Troubleshooting

For troubleshooting, refer to the Connectivity Issues section.

Snapshot Upload from Controller to Remote Path Using the NX-OS CLI

```
apic1(config)# snapshot upload policy-name
apic1(config-upload)#
 file    Snapshot file name
no      Negate a command or set its defaults
remote  Set the remote path configuration will get uploaded to

bash    bash shell for unix commands
end     Exit to the exec mode
```

```
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-upload)# file <file name from "show snapshot files">
apic1(config-upload)# remote path remote-path-name
apic1# trigger snapshot upload policy-name       [Executes the snapshot upload task]
```

Snapshot Download from Controller to Remote Path Using the NX-OS CLI

```
apic1(config)# snapshot download policy-name
apic1(config-download)#
 file    Snapshot file name
no      Negate a command or set its defaults
remote  Set the remote path configuration will get downloaded from

bash    bash shell for unix commands
end     Exit to the exec mode
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-download)# file < file from remote path>
apic1(config-download)# remote path remote-path-name
apic1# trigger snapshot download policy-name       [Executes the snapshot download task]
```

Snapshot Upload and Download Using the GUI

To upload a snapshot file to a remote location:

1. Right-click on the snapshot file listed in the **Config Rollbacks** pane, and select the **Upload to Remote Location option**. The **Upload snapshot to remote location** box appears.
2. Click **SUBMIT**.

To download a snapshot file from a remote location:

1. Click the import icon on the upper right side of the screen. The **Import remotely stored export archive to snapshot** box appears.
2. Enter the file name in the **File Name** field.
3. Select a remote location from the Import Source pull-down, or check the box next to **Or create a new one** to create a new remote location.
4. Click **SUBMIT**.

Snapshot Upload and Download Using the REST API

```
<configSnapshotManagerP name="policy-name" fileName="someexportfile.tgz" mode="upload|download"
adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configSnapshotManagerP>
```

Rollback

The **configRollbackP** policy enables you to undo the changes made between two snapshots, effectively rolling back any configuration changes that were made to the snapshot that was saved earlier. When the policy is triggered, objects are processed as follows:

- Deleted MOs are recreated
- Created MOs are deleted

- Modified MOs are reverted

Note

- The rollback feature only operates on snapshots.
- Remote archives are not supported directly. However, you can turn a remotely saved export into a snapshot using the snapshot manager policy (configSnapshotMgrP). For more information, see the Snapshot Manager Policy, on page 19
- The configRollbackP policy does not require a remote path configuration. If a remote path is provided, it will be ignored.

Rollback Workflow

The policy snapshotOneDN and snapshotTwoDn fields must be set with the first snapshot (S1) preceding snapshot two (S2). When triggered, the snapshots are extracted and analyzed to calculate and apply the differences between the snapshots.

The MOs are handled as follows:

- MOs are present in S1 but not present in S2 — These MOs were deleted before S2. The rollback will recreate these MOs.
- MOs are present in S2 but not present in S1 — These MOs were created after S1. The rollback will delete these MOs under the following circumstances:
  - These MOs were not modified after S2 was taken.
  - No MO descendants were created or modified after S2 was taken.
- MOs are present in both S1 and S2 but with different property values — If the property was modified to a different value after S2 was taken, the property is left as is. Otherwise, the rollback will revert these properties to S1.

The rollback feature also generates a diff file that contains the confiuration generated as a result of these calculations. Applying this configuration is the last step of the rollback process. The content of this file can be retrieved through a special REST API called readiff: apichost/mqapi2/snapshots.readiff.xml?jobdn=SNAPSHOT_JOB_DN.

Rollback, which is difficult to predict, also has a preview mode (set preview to true), which prevents rollback from making any actual changes. It simply calculates and generates the diff file, allowing you to preview what exactly is going to happen once the rollback is actually performed.

Diff Tool

Another special REST API is available, which provides diff functionality between two snapshots: apichost/mqapi2/snapshots.diff.xml?s1dn=SNAPSHOT_ONE_DN&s2dn=SNAPSHOT_TWO_DN.

Sample Configuration Using the NX-OS Style CLI

This example shows how to configure and execute a rollback using the NX-OS Style CLI:

```
apic1# show snapshot files
File    : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created : 2015-11-21T01:00:21.167+00:00
Root    :
Size    : 22926

File    : ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
Created : 2015-11-21T09:00:24.025+00:00
Root    :
Size    : 23588
```

```
apic1# configure
apic1(config)# snapshot rollback myRollbackPolicy
apic1(config-rollback)# first-file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-rollback)# second-file ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
apic1(config-rollback)# preview
apic1(config-rollback)# end
apic1# trigger snapshot rollback myRollbackPolicy
```

Sample Configuration Using the GUI

This example shows how to configure and execute a rollback using the GUI:

1. On the menu bar, click the **Admin** tab.
2. Click **Config Rollbacks**, located under the Admin tab.
3. Select the first configuration file from the **Config Rollbacks** list (in the left-side pane).
4. Select the second configuration file in the **Configuration for selected snapshot** pane (in the right-side pane).
5. Click the **Compare with previous snapshot** drop-down menu (at the bottom of the right-side pane), then select the second configuration file from that list. A diff file is then generated so that you can compare the differences between the two snapshots.

Note After the file generates, there is an option to undo these changes.

Sample Configuration Using the REST API

This example shows how to configure and execute a rollback using the REST API:

```
<configRollbackP name="policy-name" snapshotOneDn="dn/of/snapshot/one" snapshotOneDn="dn/of/snapshot/two"
preview="false" adminSt="triggered" />
```

Applying show running config Output to Another Cisco APIC

About Import and Export Configurations

The **import config** and **export config** commands enable you to apply the **show running config** output to another Cisco APIC. This section contains the guidelines for these commands and demonstrates how the commands are executed.

Import and Export Configuration Guidelines and Limitations

This section explains the guidelines and limitations for the **export config** and **import config** commands.

- Passwords and other encrypted data are not included in the configuration file.
- Some REST API configurations may not be compatible with CLI configurations; this may cause errors when applying a configuration file to a Cisco APIC.
- Some features require configurations to be in a specific order. These configurations are validated when performed through the CLI. Configurations through the REST API, however, are not validated and may cause errors when running the imported file due to missing configurations.
- Interactive commands are prefixed with a "#" and ignored when running the configuration file.

Exporting a CLI Configuration

This procedure shows how to export a configuration to a text file.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `dev4-ifc1# configure`	Enters configuration mode.
Step 2	**leaf** *ID* **Example:** `dev4-ifc1(config)# leaf 101`	Identifies the leaf with the configuration to be exported.
Step 3	**interface ethernet** *slot/port* **Example:** `dev4-ifc1(config-leaf)# interface ethernet 1/34`	Identifies the slot number and port number for an existing Ethernet interface.
Step 4	**export-config** *result-file-name* **Example:** `dev4-ifc1(config-leaf-if)# export-config /tmp/showRunnLeaf101.txt`	Exports the configuration to a specified file name.

Example

This example shows how to configure export-config.

```
dev4-ifc1# config
dev4-ifc1(config)# leaf 101
dev4-ifc1(config-leaf)# interface ethernet 1/34
dev4-ifc1(config-leaf-if)# export-config /tmp/showRunnLeaf101.txt
dev4-ifc1(config-leaf-if)# cat /tmp/showRunnLeaf101.txt
config
# Command: show running-config leaf 101 interface ethernet 1 / 34
# Time: Fri Sep 23 16:03:48 2016
  leaf 101
    interface ethernet 1/34
      switchport trunk allowed vlan 602 tenant t1 external-svi l3out l3ext1sub1
      exit
    exit
dev4-ifc1(config-leaf-if)#
```

Importing a CLI Configuration

This procedure shows how to import a configuration from a text file.

Procedure

	Command or Action	Purpose
Step 1	**import-config** *file-name*	

	Command or Action	Purpose

	Example:
	`dev4-ifc1(config-tenant)# import-config /tmp/showRunnLeaf101.txt`
	`config`
	`# Command: show running-config leaf 101 interface ethernet 1 / 34`
	`# Time: Fri Sep 23 16:03:48 2016`
	`leaf 101`
	`interface ethernet 1/34`
	`switchport trunk allowed vlan 602 tenant t1`
	`external-svi l3out l3ext1sub1`
	`exit`
	`exit`
	`dev4-ifc1(config)#`	

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2015–2021 Cisco Systems, Inc. All rights reserved.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.