Using the Scripting Interface

This chapter describes the scripting interface that ACS 5.8.1 provides to perform bulk operations on ACS objects using the Import and Export features.

ACS provides the import and export functionalities through the web interface (graphical user interface) as well as the CLI. ACS exposes these functionalities through the CLI to enable you to create custom shell scripts for bulk operations on ACS objects. The **import-data** command allows you to:

- Add ACS objects
- Update ACS objects
- Delete ACS objects

The import and export functionalities in ACS 5.8.1 allow you to perform bulk operations such as Create, Update, and Delete on ACS objects and provide a migration path for customers migrating from ACS 4.*x* releases to ACS 5.8.1.

You can integrate ACS with any of your repositories and import data into ACS through automated scripts, using the Import and Export features. You can also encrypt the .csv file before you transfer the file for additional security, or, optionally, use Secure File Transfer Protocol (SFTP).

You can create a scheduled command that looks for a file with a fixed name in the repository to perform bulk operations. This option provides the functionality that was available in ACS 4.x releases.

ACS processes the import and export requests in a queue. Only one process can run at a time. When you use the ACS web interface for importing and exporting, you cannot manually control the queue.

ACS processes the queue in sequence. However, you can use the CLI to manage the import and export processes in ACS. The ACS CLI allows you to view the status of the queue and terminate the processes that are in the queue.

This chapter contains the following sections:

- Understanding Import and Export in ACS, page 1
- Supported ACS Objects, page 4
- Creating Import Files, page 6
- Using Shell Scripts to Perform Bulk Operations, page 10

Understanding Import and Export in ACS

You can use the import functionality in ACS to add, update, or delete multiple ACS objects at the same time. ACS uses a comma-separated values (CSV) file to perform these bulk operations. This *.csv* file is called an import file.

ACS provides a separate *.csv* template for Add, Update, and Delete operations for each ACS object. The first record in the .csv file is the header record from the template that contains column (field) names. You must download these templates from the ACS web interface. The header record from the template must be included in the first row of any .csv file that you import.

You cannot use the same template to import all ACS objects. You must download the template that is designed for each ACS object and use the corresponding template while importing the objects.

You can use the export functionality to create a *.csv* file that contains all the records of a particular object type that are available in the ACS internal store.

You must have CLI administrator-level access to perform import and export operations. Additionally:

- To import ACS configuration data, you need CRUD permissions for the specific configuration object.
- To export data to a remote repository, you need read permission for the specific configuration object.

This section contains:

- Importing ACS Objects Through the CLI, page 2
- Exporting ACS Objects Through the CLI, page 3
- Viewing the Status of Import and Export Processes, page 4
- Terminating Import and Export Processes, page 4

Importing ACS Objects Through the CLI

You can import ACS objects from the ACS Configuration mode. You use the **import-data** command to perform the Import operation. This command takes the following arguments:

- Name of the remote repository where the import file resides. See Creating Import Files, page 6, for information on how to create the import file.
- Name of the import file.
- Type of ACS object that the import file contains.

ACS obtains the *.csv* file from the remote repository and processes the file. You can query ACS for the status of the import process using the **import-export-status** command. After the import process is complete, ACS generates a status file in the remote repository that includes any errors that ACS identified during this process.

For additional security during the import process, you have the option of encrypting the import file and using a secured remote repository for the import operation.

Also, the import process sometimes can run into errors. You can specify whether you want to terminate the import process or continue it until it is complete.

Note: If you choose to use a secured remote repository for import, you must specify SFTP as the *repository* value.

For example, to add internal user records to an existing identity store, from the ACS CLI, enter:

```
import-data add user repository file-name result-file-name {abort-on-error | cont-on-error} {full | none |
only-sec-repo | only-sec-files} secret-phrase
```

Syntax Description

repository—Name of the remote repository from which to import the ACS objects, in this case, the internal users.

file-name—Name of the import file in the remote repository.

result-file-name—Name of the file that contains the results of the import operation. This file is available in the remote repository when the import process completes or is terminated.

abort-on-error—Aborts the import operation if an error occurs during the import process.

cont-on-error—Ignores any errors that occur during the import process and continues to import the rest of the object.

full—Encrypts the import file using the GNU Privacy Guard (GPG) encryption mechanism and uses secured remote repository to import the file. If you specify the security type as **full**, you must specify SFTP as the repository value.

none—Neither encrypts the import file nor uses the secured remote repository for import.

secret phrase—Provide the secret phrase to decrypt the import file. If you specify the security type as **full** or **only-sec-files**, you must specify the secret phrase.

only-sec-repo—Uses the secured remote repository to import the file. If you specify the security type as **only-sec-repo**, you must specify SFTP as the repository value.

only-sec-files–Encrypts the import file using GPG encryption mechanism.

For more information on the **import-data** command, see CLI Reference Guide for Cisco Secure Access Control System 5.8.1.

Exporting ACS Objects Through the CLI

You can export a list of ACS objects in a *.csv* file from ACS to your local drive. You can perform this operation from the ACS Configuration mode, using the **export-data** command. This command takes the following arguments:

- Object type to be exported.
- Name of the remote repository to which the *.csv* file should be downloaded after the export process is complete.

When ACS processes your export request, you can enter a command to query the progress of the export. After the export process is complete, the *.csv* file that is available in your remote repository should contain all the object records that exist in the ACS internal store.

Note: When you export ACS objects through the web interface, use the available filters to export a subset of the records.

For additional security during the export process, you have the option of encrypting the export file and using a secured remote repository for the export operation.

Note: If you choose to use a secured remote repository for export, you must specify SFTP as the repository value.

For example, to export internal user records, from the ACS CLI, enter:

export-data user *repository file-name result-file-name* **{full** | **none** | **only-sec-repo** | **only-sec-files}** *secret-phrase*

Syntax Description

repository—Name of the remote repository to which to export the ACS objects, in this case, the internal users.

file-name—Name of the export file in the remote repository.

result-file-name—Name of the file that contains the results of the export operation. This file is available in the remote repository when the export process completes.

full—Encrypts the export file using the GPG encryption mechanism and uses secured remote repository to export the file. If you specify the security type as **full**, you must specify SFTP as the repository value.

none—Neither encrypts the export file nor uses the secured remote repository for export.

secret phrase—Provide a secret phrase to encrypt the export file. If you specify the security type as **full** or **only-sec-files**, you must specify the secret phrase.

only-sec-repo—Uses the secured remote repository to export the file. If you specify the security type as only-sec-repo, you must specify SFTP as the repository value.

only-sec-files—Encrypts the export file using GPG encryption mechanism.

For more information on the **export-data** command, see CLI Reference Guide for Cisco Secure Access Control System 5.8.1.

Viewing the Status of Import and Export Processes

You can view the status of the import and export processes in ACS using the **import-export-status** command. Use this command to view the status of running import and export processes and to verify whether there are any pending processes.

You must run the **import-export-status** command from the ACS Configuration mode. Any user, irrespective of role, can issue this command.

import-export-status {*current* | *all* | **id** *id*}

Syntax Description

current–Displays the status of the currently running processes.

all–Displays the status of all the import and export processes, including any pending processes.

id–Displays the import or export status, based on a particular process that is specified by the process ID.

For more information on the **import-export-status** command, see CLI Reference Guide for Cisco Secure Access Control System 5.8.1.

Terminating Import and Export Processes

You can use the **import-export-abort** command to terminate all import and export processes, or process that are currently running or queued. You must run the **import-export-abort** command from the ACS Configuration mode.

Only the super administrator can simultaneously terminate a running process and all pending import and export processes. However, a user who owns a particular import or export process can terminate that particular process by using the process ID, or by stopping the process when it is running.

import-export-abort {*running* | *all* | **id** *id*}

Syntax Description

current–Aborts any import or export process that is running currently.

all–Aborts all the import and export processes in the queue.

id–Aborts the import or export process, based on the process ID that you specify.

For more information on the **import-export-abort** command, see CLI Reference Guide for Cisco Secure Access Control System 5.8.1.

Supported ACS Objects

While ACS 5.8.1 allows you to perform bulk operations (Add, Update, Delete) on ACS objects using the import functionality, you cannot import all ACS objects. The import functionality in ACS 5.8.1 supports the following ACS objects:

- Users
- Hosts
- Network Devices
- Identity Groups

- NDGs
- Downloadable ACLs
- Command Sets

Table 1 on page 5 lists the ACS objects, their properties, and the property data types.

Table 1 ACS Objects – Property Names and Data Types

Property Name	Property Data Type
Object Type: User	
Username	(Required in create, edit, and delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Enabled	(Required in create) Boolean.
Change Password	(Required in create) Boolean.
Password	(Required in create) String. Maximum length is 32 characters. Not available in Export.
Enable Password	(Optional) String. Maximum length is 32 characters.
User Identity Group	(Optional) String. Maximum length is 256 characters.
List of attributes	(Optional) String and other data types.
Object Type: Hosts	
MAC address	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Enabled	(Optional) Boolean.
Host Identity Group	(Optional) String. Maximum length is 256 characters.
List of attributes	(Optional) String.
Object Type: Network Device	
Name	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Subnet	(Required in create) String.
Support RADIUS	(Required in create) Boolean.
RADIUS secret	(Optional) String. Maximum length is 32 characters.
Support TACACS	(Required in create) Boolean.
TACACS secret	(Optional) String. Maximum length is 32 characters.
Single connect	(Optional) Boolean.
Legacy TACACS	(Optional) Boolean.
Support CTS	(Required in create) Boolean.
CTS Identity	(Optional) String. Maximum length is 32 characters.
CTS trusted	(Optional) Boolean.
Password	(Optional) String. Maximum length is 32 characters.
sgACLTTL	(Optional) Integer.
peerAZNTTL	(Optional) Integer.
envDataTTL	(Optional) Integer.

Table 1 ACS Objects - Property Names and Data Types (continued)

Property Name	Property Data Type
Session timeout	(Optional) Integer.
List of NDG names	(Optional) String.
Object Type: Identity Group	
Name	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Object Type: NDG	
Name	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Object Type: Downloadable ACLs	
Name	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Content	(Required in create, edit, delete) String. Maximum length is 1024 characters.
Object Type: Command Set	
Name	(Required in create, edit, delete) String. Maximum length is 64 characters.
Description	(Optional) String. Maximum length is 1024 characters.
Commands (in the form of *grant*:*command*:*arguments*)	(Optional) String. **Note:** This is a list with semicolons used as separators (:) between the values that you supply for *grant*.

Fields that are optional can be left empty and ACS substitutes the default values for those fields.

For example, when fields that are related to a hierarchy are left blank, ACS assigns the value of the root node in the hierarchy. For network devices, if TrustSec is enabled, all related configuration fields are set to default values.

Note: You can export Administrators from the CLI as well as the ACS user interface. To export Administrators from the ACS user interface, use the export button available in the Administrator Accounts page. To export Administrators from the CLI, use the **export-data** command. For more information on the **export-data** command, see CLI Reference Guide for Cisco Secure Access Control System 5.8.1.

Note: You can export message catalog from the CLI as well as the ACS user interface. To export message catalog from the ACS user interface, use the export button available in the **System Administration > Configuration > Log Configuration > Log Message Catalog** page. To export message catalog from the CLI, use the **export-message-catalog** command.

Creating Import Files

This section describes how to create the *csv* file for performing bulk operations on ACS objects. You can download the appropriate template for each of the objects. This section contains the following:

- Downloading the Template from the Web Interface, page 7
- Understanding the CSV Templates, page 8
- Creating the Import File, page 8

Downloading the Template from the Web Interface

Before you can create the import file, you must download the import file templates from the ACS web interface.

To download the import file templates for adding internal users:

1. Log into the ACS 5.8.1 web interface.
2. Choose **Users and Identity Stores > Internal Identity Stores > Users**.

   The Users page appears.
3. Click **File Operations**.

   The File Operations wizard appears.
4. Choose any one of the following:

- Add—Adds users to the existing list. This option does not modify the existing list. Instead, it performs an append operation.
- Update—Updates the existing internal user list.
- Delete—Deletes the list of users in the import file from the internal identity store.

5. Click **Next**.

   The Template page appears.
6. Click **Download Add Template**.
7. Click **Save** to save the template to your local disk.

The following list gives you the location from which you can get the appropriate template for each of the objects:

- User—**Users and Identity Stores > Internal Identity Stores > Users**
- Hosts—**Users and Identity Stores > Internal Identity Stores > Hosts**
- Network Device—**Network Resources > Network Devices and AAA Clients**
- Identity Group—**Users and Identity Stores > Identity Groups**
- NDG
  - Location—**Network Resources > Network Device Groups > Location**
  - Device Type—**Network Resources > Network Device Groups > Device Type**
- Downloadable ACLs—**Policy Elements > Authorization and Permissions > Named Permission Objects > Downloadable ACLs**
- Command Set—**Policy Elements > Authorization and Permissions > Device Administration > Command Sets**

Follow the procedure described in this section to download the appropriate template for your object.

Understanding the CSV Templates

You can open your CSV template in Microsoft Excel or any other spreadsheet application and save the template to your local disk as a *.csv* file. The *.csv* template contains a header row that lists the properties of the corresponding ACS object.

For example, the internal user Add template contains the fields described in Table 2 on page 8:

Table 2 Internal User Add Template

Header Field	Description
name:String(64):Required	Username of the user.
description:String(1024)	Description of the user.
enabled:Boolean(true,false):Required	Boolean field that indicates whether the user must be enabled or disabled.
changePassword:Boolean(true,false): Required	Boolean field that indicates whether the user must change password on first login.
password:String(32):Required	Password of the user.
enablePassword:String(32)	Enable password of the user.
UserIdentityGroup:String(256)	Identity group to which the user belongs.
All the user attributes that you have specified would appear here.	

Each row of the *.csv* file corresponds to one internal user record. You must enter the values into the *.csv* file and save it before you can import the users into ACS. See Creating the Import File, page 8 for more information on how to create the import file.

This example is based on the internal user Add template. For the other ACS object templates, the header row contains the properties described in Table 1 on page 5 for that object.

Creating the Import File

After you download the import file template to your local disk, enter the records that you want to import into ACS in the format specified in the template. After you enter all records into the *.csv* file, you can proceed with the import function. The import process involves the following:

- Adding Records to the ACS Internal Store, page 8
- Updating the Records in the ACS Internal Store, page 9
- Deleting Records from the ACS Internal Store, page 10

Adding Records to the ACS Internal Store

When you add records to the ACS internal store, you add the records to the existing list. This is an append operation, in which the records in the *.csv* file are added to the list that exists in ACS.

To add internal user records to the Add template:

1. Download the internal user Add template. See Downloading the Template from the Web Interface, page 7 for more information.
2. Open the internal user Add template in Microsoft Excel or any other spreadsheet application. See Table 1 on page 5 for a description of the fields in the header row of the template.
3. Enter the internal user information. Each row of the .csv template corresponds to one user record.

   Figure 1 on page 9 Figure 1 on page 9 shows a sample Add Users import file.

Figure 1 Add Users – Import File

	A	B	C	D	E	F	G	H	I	J	K
1	name:String(64):Re	description:Str	enabled:	changePassw	password:Stri	enablePasswo	UserIdentityGroup:Str	attr-Real Name	attr-Description:String(256)		
2	John		TRUE	FALSE	1234		All Groups:SanJose				
3	Kenneth		TRUE	FALSE	1235		All Groups:SanJose				
4	Abraham		TRUE	FALSE	1236		All Groups:Texas				
5	Kelly		TRUE	FALSE	1237		All Groups:Texas				
6	Sandra		TRUE	FALSE	1238		All Groups:Florida				
7	Nilofer		TRUE	FALSE	1239		All Groups:Florida				
8	James		TRUE	FALSE	1240		All Groups:SanJose				
9	Albert		TRUE	FALSE	1241		All Groups:Florida				
10	Kevin		TRUE	FALSE	1242		All Groups:Florida				
11	Samantha		TRUE	FALSE	1243		All Groups:Texas				

252953

4. Save the add users import file to your local disk.

Updating the Records in the ACS Internal Store

When you update the records in the ACS store, the import process overwrites the existing records in the internal store with the records from the *.csv* file. This operation replaces the records that exist in ACS with the records from the *.csv* files.

The Update operation is similar to the Add operation except for one additional column that you can add to the Update templates.

The Update template can contain an Updated Name column for internal users and other ACS objects, and an Updated MAC address column for the internal hosts. The name shown in the Updated Name column replaces the name in the ACS identity store.

Instead of downloading the update template for each of the ACS objects, you can use the export file of that object, retain the header row, and update the data to create your updated .csv file.

To add an updated name or MAC address to the ACS objects, you must download and use the particular update template. Also, for the NDGs, the export template contains only the NDG name, so in order to update any other property, you must download and use the NDG update template.

Figure 2 on page 9 shows a sample import file that updates existing user records.

Figure 2 Update Users–Import File

	A	B	C	D	E	F	G	H	I	J	K	L
1	name:Strir	Updated name:Strir	description	enabled:B	changePas	password:S	enablePas	UserIdentit	attr-Real N	attr-Description:String(256)		
2	John	Mark		TRUE	FALSE	1234		All Groups:SanJose				
3	Kenneth	David		TRUE	FALSE	1235		All Groups:SanJose				
4	Abraham	Jamie		TRUE	FALSE	1236		All Groups:Texas				
5	Kelly	Lucy		TRUE	FALSE	1237		All Groups:Texas				
6	Sandra	Tina		TRUE	FALSE	1238		All Groups:Florida				
7	Nilofer	William		TRUE	FALSE	1239		All Groups:Florida				
8	James	Frank		TRUE	FALSE	1240		All Groups:SanJose				
9	Albert	George		TRUE	FALSE	1241		All Groups:Florida				
10	Kevin	Paul		TRUE	FALSE	1242		All Groups:Florida				
11	Samantha	Patrick		TRUE	FALSE	abcd		All Groups:Texas				

252971

Note: The second column, Updated name, is the additional column that you can add to the Update template. Also, the password value and the enabled password value are not mandatory in the case of an update operation for the user object.

Deleting Records from the ACS Internal Store

You can use this option to delete a subset of records from the ACS internal store. The records that are present in the *.csv* file that you import are deleted from the ACS internal store. The Delete template contains only the key column to identify the records that must be deleted.

For example, to delete a set of internal users from the ACS internal identity store, download the internal user Delete template and add the list of users that you want to delete to this Import file. Figure 3 on page 10 shows a sample Import file that deletes internal user records.

Note: To delete all users, you can export all users and then use the export file as your import file to delete users.

Figure 3 Delete Users – Import File

Using Shell Scripts to Perform Bulk Operations

You can write custom shell scripts that use the import and export CLI commands to perform bulk operations. The ACS web interface provides a sample Python script. To download this sample script:

1. Log into the ACS web interface.
2. Choose **System Administration > Downloads > Scripts**.

The downloadable package consists of:

- Python module, Pexpect
- Python script
- ReadMe—Contains installation instructions

Note: You must have Python software to run this script.

Sample Shell Script

```
import pexpect

# Create connection to a specific IP using 'admin' username
connector = pexpect.spawn('ssh admin@1.2.3.4')
connector.expect('.ssword:*')
# Enter password
connector.sendline('defaultPass')
connector.expect('.$')
# Defining a repository that point to the localdisc
connector.sendline('configure')
connector.expect('.$')
connector.sendline('repository localRepo')
```

```
connector.expect('.$')
connector.sendline('url disk:/')
connector.expect('.$')
connector.sendline('exit')
connector.expect('.$')
connector.sendline('exit')
connector.expect('.$')
# Saving the repository
connector.sendline('write memory')
connector.expect('.$')
# Going into acs-config mode
connector.sendline('acs-config')
connector.expect('.ername:*')
# Enter acs admin username
connector.sendline('acsadmin')
connector.expect('.ssword:*')
# Enter acs admin password
connector.sendline('1111')
connector.expect('.config-acs*')
connector.sendline('import-data add device local device.csv device_res.csv cont-on-error none')   #
Performing the import command
connector.expect('.$')
# Exit acs-config mode
connector.sendline('exit')
connector.expect('.$')
# Exit ssh mode
connector.sendline('exit')
```