



## CSUtil Database Utility

---

This appendix details the Cisco Secure Access Control Server (ACS) for Windows Server command-line utility, CSUtil.exe. Among its several functions, CSUtil.exe enables you to add, change, and delete users from a colon-delimited text file. You can also use the utility to add and delete AAA client configurations.



### Note

---

You can accomplish similar tasks using the ACS System Backup, ACS System Restore, Database Replication, and RDBMS Synchronization features. For more information on these features, see [Chapter 9, “System Configuration: Advanced”](#).

---

This chapter contains the following topics:

- [Location of CSUtil.exe and Related Files, page D-2](#)
- [CSUtil.exe Syntax, page D-2](#)
- [CSUtil.exe Options, page D-3](#)
- [Displaying Command-Line Syntax, page D-5](#)
- [Backing Up Cisco Secure ACS with CSUtil.exe, page D-6](#)
- [Restoring Cisco Secure ACS with CSUtil.exe, page D-7](#)
- [Creating a CiscoSecure User Database, page D-8](#)
- [Creating a Cisco Secure ACS Database Dump File, page D-10](#)
- [Loading the Cisco Secure ACS Database from a Dump File, page D-11](#)
- [Compacting the CiscoSecure User Database, page D-12](#)
- [User and AAA Client Import Option, page D-14](#)

- [Exporting User List to a Text File, page D-24](#)
- [Exporting Group Information to a Text File, page D-25](#)
- [Exporting Registry Information to a Text File, page D-26](#)
- [Decoding Error Numbers, page D-27](#)
- [Recalculating CRC Values, page D-28](#)
- [User-Defined RADIUS Vendors and VSA Sets, page D-28](#)
- [PAC File Generation, page D-40](#)
- [Posture Validation Attributes, page D-44](#)

## Location of CSUtil.exe and Related Files

When you install Cisco Secure ACS in the default location, CSUtil.exe is located in the following directory:

```
C:\Program Files\CiscoSecure ACS vX.X\Utils
```

where *X.X* is the version of your Cisco Secure ACS software. Regardless of where you install Cisco Secure ACS, CSUtil.exe is located in the `Utils` directory.

Files generated by or accessed by CSUtil.exe are also located in the `Utils` directory.

## CSUtil.exe Syntax

The syntax for the CSUtil.exe command is as follows:

```
CSUtil.exe [-q] [-c] [-d] [-g] [-i filename] [[-p] -l filename] [-e -number]
[-b filename] [-r filename] [-f] [-n] [-u] [-x] [-y] [-listUDV] [-addUDV slot filename]
[-delUDV slot] [-t -filepath full filepath] [-passwd password] {-a | -g group number |
-u username | -f user list filepath}] [-addAVP filename]
[-delAVP vendor-ID application-ID attribute-ID] [-dumpAVP filename]
```

**Note**

Most CSUtil.exe options require that you stop the CSAuth service. While the CSAuth service is stopped, Cisco Secure ACS does not authenticate users. To determine if an option requires that you stop CSAuth, refer to the detailed topics about the option. For a list of options and references to the detailed topics about each option, see [CSUtil.exe Options, page D-3](#).

You can combine many of the options in a single use of CSUtil.exe. If you are new to using CSUtil.exe, we recommend performing only one option at a time, with the exception of those options, such as `-p`, that must be used in conjunction with other options.

Experienced CSUtil.exe users may find it useful to combine CSUtil.exe options, such as in the following example, which would first import AAA client configurations and then generate a dump of all Cisco Secure ACS internal data:

```
CSUtil.exe -i newnases.txt -d
```

## CSUtil.exe Options

CSUtil.exe can perform several actions. The options, listed below in alphabetical order, are detailed in later sections of this chapter.

- **-b**—Backup system to a specified filename. For more information about this option, see [Backing Up Cisco Secure ACS with CSUtil.exe, page D-6](#).
- **-c**—Recalculate database CRC values. For more information about this option, see [Recalculating CRC Values, page D-28](#).
- **-d**—Export all Cisco Secure ACS internal data to a file named `dump.txt`. Using this option requires that you stop the CSAuth service. For more information about this option, see [Creating a Cisco Secure ACS Database Dump File, page D-10](#).
- **-e**—Decode internal Cisco Secure ACS error numbers to ASCII message. For more information about this option, see [Decoding Error Numbers, page D-27](#).
- **-g**—Export group information to a file named `groups.txt`. Using this option requires that you stop the CSAuth service. For more information about this option, see [Exporting Group Information to a Text File, page D-25](#).

- **-i**—Import user or AAA client information from a file named `import.txt` or a specified file. For more information about this option, see [Importing User and AAA Client Information, page D-15](#).
- **-l**—Load all Cisco Secure ACS internal data from a file named `dump.txt` or named file. Using this option requires that you stop the CSAuth service. For more information about this option, see [Loading the Cisco Secure ACS Database from a Dump File, page D-11](#).
- **-n**—Create CiscoSecure user database and index. Using this option requires that you stop the CSAuth service. For more information about this option, see [Creating a CiscoSecure User Database, page D-8](#).
- **-p**—Reset password aging counters during database load, to be used only in conjunction with the `-l` option. For more information about this option, see [Loading the Cisco Secure ACS Database from a Dump File, page D-11](#).
- **-q**—Run CSUtil.exe without confirmation prompts.
- **-r**—Restore system from a specified backup filename. For more information about this option, see [Restoring Cisco Secure ACS with CSUtil.exe, page D-7](#).
- **-t**—Generate PAC files for EAP-FAST end-user clients. For more information about this option, see [PAC File Generation, page D-40](#).
- **-u**—Export user information, sorted by group membership, to a file named `users.txt`. Using this option requires that you stop the CSAuth service. For more information about this option, see [Exporting User List to a Text File, page D-24](#).
- **-x**—Display command-line syntax. For more information about this option, see [Displaying Command-Line Syntax, page D-5](#).
- **-y**—Dump Windows Registry configuration information to a file named `setup.txt`. For more information about this option, see [Exporting Registry Information to a Text File, page D-26](#).
- **-addUDV**—Add a user-defined RADIUS vendor-specific attribute (VSA). For more information about this option, see [Adding a Custom RADIUS Vendor and VSA Set, page D-29](#).
- **-delUDV**—Delete a user-defined RADIUS VSA. For more information about this option, see [Deleting a Custom RADIUS Vendor and VSA Set, page D-31](#).

- **-listUDV**—List all user-defined RADIUS VSAs currently defined in Cisco Secure ACS. For more information about this option, see [Listing Custom RADIUS Vendors, page D-32](#).
- **-addAVP**—Add or modify a posture validation attribute. For more information about this option, see [Importing Posture Validation Attribute Definitions, page D-49](#).
- **-delAVP**—Delete a posture validation attribute. For more information about this option, see [Deleting a Posture Validation Attribute Definition, page D-51](#).
- **-dumpAVP**—Export all posture validation attributes. For more information about this option, see [Exporting Posture Validation Attribute Definitions, page D-48](#).

## Displaying Command-Line Syntax

CSUtil.exe displays command-line syntax for any one of the following reasons:

- The `-x` option is included in the CSUtil.exe command.
- No options are included with the CSUtil.exe command.
- Incorrect syntax is used with the CSUtil.exe command.

For more information about CSUtil.exe syntax, see [CSUtil.exe Syntax, page D-2](#).

To display command-line syntax for CSUtil.exe, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** Type:

```
CSUtil.exe -x
```

Press **Enter**.

CSUtil.exe displays its command-line syntax.

---

# Backing Up Cisco Secure ACS with CSUtil.exe

You can use the `-b` option to create a system backup of all Cisco Secure ACS internal data. The resulting backup file has the same data as the backup files produced by the ACS Backup feature found in the HTML interface. For more information about the ACS Backup feature, see [Cisco Secure ACS Backup, page 8-9](#).

**Note**

---

During the backup, all services are automatically stopped and restarted. No users are authenticated while the backup is occurring.

---

To back up Cisco Secure ACS with CSUtil.exe, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** Type:  
`CSUtil.exe -b filename`

where *filename* is the name of the backup file. Press **Enter**.

CSUtil.exe displays a confirmation prompt.

**Step 3** To confirm that you want to perform a backup and to halt all Cisco Secure ACS services during the backup, type **Y** and press **Enter**.

CSUtil.exe generates a complete backup of all Cisco Secure ACS internal data, including user accounts and system configuration. This process may take a few minutes.

**Note**

---

CSUtil.exe displays the error message “Backup Failed” when it attempts to back up components of Cisco Secure ACS that are empty, such as when no administrator accounts exist. These apply only to components that are empty, not to the overall success or failure of the backup.

---

# Restoring Cisco Secure ACS with CSUtil.exe

You can use the `-r` option to restore all Cisco Secure ACS internal data. The backup file from which you restore Cisco Secure ACS can be one generated by the CSUtil.exe `-b` option or by the ACS Backup feature in the HTML interface.

Cisco Secure ACS backup files contain two types of data:

- User and group data.
- System configuration.

You can restore either user and group data or system configuration, or both. For more information about the ACS Backup feature, see [Cisco Secure ACS Backup, page 8-9](#).



## Note

---

During the restoration, all services are automatically stopped and restarted. No users are authenticated while the restoration is occurring.

---

To restore Cisco Secure ACS with CSUtil.exe, follow these steps:

- 
- Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).
- Step 2** Perform one of the following:
- To restore all data (user and group data, and system configuration), type:  

```
CSUtil.exe -r all filename
```

where *filename* is the name of the backup file. Press **Enter**.
  - To restore only user and group data, type:  

```
CSUtil.exe -r users filename
```

where *filename* is the name of the backup file. Press **Enter**.

- To restore only the system configuration, type:

```
CSUtil.exe -r config filename
```

where *filename* is the name of the backup file. Press **Enter**.

CSUtil.exe displays a confirmation prompt.

- Step 3** To confirm that you want to perform a restoration and to halt all Cisco Secure ACS services during the restoration, type **Y** and press **Enter**.

CSUtil.exe restores the specified portions of your Cisco Secure ACS data. This process may take a few minutes.




---

**Note** If the backup file is missing a database component, CSUtil.exe displays an error message. Such an error message applies only to the restoration of the missing component. The absence of a database component in a backup is usually intentional and indicates that the component was empty in Cisco Secure ACS at the time the backup was created.

---

## Creating a CiscoSecure User Database

You can use the `-n` option to create a CiscoSecure user database.



**Note**

---

Using the `-n` option requires that you stop the CSAuth service. While CSAuth is stopped, no users are authenticated.

---



**Caution**

---

Using the `-n` option erases all user information in the CiscoSecure user database. Unless you have a current backup or dump of your CiscoSecure user database, all user accounts are lost when you use this option.

---

To create a CiscoSecure user database, follow these steps:

- 
- Step 1** If you have not performed a backup or dump of the CiscoSecure user database, do so now before proceeding. For more information about backing up the database, see [Backing Up Cisco Secure ACS with CSUtil.exe, page D-6](#). For more information about performing a dump of the database, see [Creating a Cisco Secure ACS Database Dump File, page D-10](#).
- Step 2** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).
- Step 3** If the CSAuth service is running, type:
- ```
net stop csauth
```
- and press **Enter**.
- The CSAuth service stops.
- Step 4** Type:
- ```
CSUtil.exe -n
```
- and press **Enter**.
- CSUtil.exe displays a confirmation prompt.
- Step 5** To confirm that you want to initialize the CiscoSecure user database, type **Y** and press **Enter**.
- The CiscoSecure user database is initialized. This process may take a few minutes.
- Step 6** To resume user authentication, type:
- ```
net start csauth
```
- and press **Enter**.
-

# Creating a Cisco Secure ACS Database Dump File

You can use the `-d` option to dump all contents of the CiscoSecure user database into a text file. This provides a thorough and compressible backup of all Cisco Secure ACS internal data.

Using the `-l` option, you can reload the Cisco Secure ACS internal data from a dump file created by the `-d` option. For more information about the `-l` option, see [Loading the Cisco Secure ACS Database from a Dump File, page D-11](#).

**Note**

---

Using the `-d` option requires that you stop the CSAuth service. While CSAuth is stopped, no users are authenticated.

---

To dump all Cisco Secure ACS internal data into a text file, follow these steps:

- 
- Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).
- Step 2** If the CSAuth service is running, type:
- ```
net stop csauth
```
- and press **Enter**.
- The CSAuth service stops.
- Step 3** Type:
- ```
CSUtil.exe -d
```
- and press **Enter**.
- CSUtil.exe displays a confirmation prompt.
- Step 4** To confirm that you want to dump all Cisco Secure ACS internal data into `dump.txt`, type **Y** and press **Enter**.
- CSUtil.exe creates the `dump.txt` file. This process may take a few minutes.

**Step 5** To resume user authentication, type:

```
net start csauth
```

and press **Enter**.

---

## Loading the Cisco Secure ACS Database from a Dump File

You can use the `-l` option to overwrite all Cisco Secure ACS internal data from a dump text file. This option replaces the existing all Cisco Secure ACS internal data with the data in the dump text file. In effect, the `-l` option initializes all Cisco Secure ACS internal data before loading it from the dump text file. Dump text files are created using the `-d` option. While the `-d` option only produces dump text files that are named `dump.txt`, the `-l` option allows for loading renamed dump files. For more information about creating dump text files, see [Creating a Cisco Secure ACS Database Dump File, page D-10](#).

You can use the `-p` option in conjunction with the `-l` option to reset password-aging counters.



### Note

Using the `-l` option requires that you stop the CSAuth service. While CSAuth is stopped, no users are authenticated.

---

To load all Cisco Secure ACS internal data from a text file, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** If the CSAuth service is running, type:

```
net stop csauth
```

and press **Enter**.

The CSAuth service stops.

**Step 3** Type:

```
CSUtil.exe -l filename
```

where *filename* is the name of the dump file you want CSUtil.exe to use to load Cisco Secure ACS internal data. Press **Enter**.

CSUtil.exe displays a confirmation prompt for overwriting all Cisco Secure ACS internal data with the data in the dump text file.



---

**Note** Overwriting the database does not preserve any data; instead, after the overwrite, the database contains only what is specified in the dump text file.

---

**Step 4** To confirm that you want to replace all Cisco Secure ACS internal data, type **Y** and press **Enter**.

CSUtil.exe initializes all Cisco Secure ACS internal data, and then loads Cisco Secure ACS with the information in the dump file specified. This process may take a few minutes.

**Step 5** To resume user authentication, type:

```
net start csauth
```

and press **Enter**.

---

## Compacting the CiscoSecure User Database

Like many relational databases, the CiscoSecure user database handles the deletion of records by marking deleted records as deleted but not removing the records from the database. Over time, your CiscoSecure user database may be substantially larger than is required by the number of users it contains. To reduce the CiscoSecure user database size, you can compact it periodically.

Compacting the CiscoSecure user database consists of using in conjunction three CSUtil.exe options:

- **-d**—Export all Cisco Secure ACS internal data to a text file named `dump.txt`.
- **-n**—Create a CiscoSecure user database and index.
- **-l**—Load all Cisco Secure ACS internal data from a text file. If you do not specify the filename, CSUtil.exe uses the default file name `dump.txt`.

Additionally, if you want to automate this process, consider using the `-q` option to suppress the confirmation prompts that otherwise appear before CSUtil.exe performs the `-n` and `-l` options.

**Note**

Compacting the CiscoSecure user database requires that you stop the CSAuth service. While CSAuth is stopped, no users are authenticated.

To compact the CiscoSecure user database, follow these steps:

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** If the CSAuth service is running, type:

```
net stop csauth
```

and press **Enter**.

The CSAuth service stops.

**Step 3** Type:

```
CSUtil.exe -d -n -l
```

Press **Enter**.

**Tip**

If you include the `-q` option in the command, CSUtil.exe does not prompt you for confirmation of initializing or loading the database.

If you do not use the `-q` option, CSUtil.exe displays a confirmation prompt for initializing the database and then for loading the database. For more information about the effects of the `-n` option, see [Creating a CiscoSecure User Database, page D-8](#). For more information about the effects of the `-l` option, see [Loading the Cisco Secure ACS Database from a Dump File, page D-11](#).

**Step 4** For each confirmation prompt that appears, type **Y** and press **Enter**.

CSUtil.exe dumps all Cisco Secure ACS internal data to `dump.txt`, initializes the CiscoSecure user database, and reloads all Cisco Secure ACS internal data from `dump.txt`. This process may take a few minutes.

**Step 5** To resume user authentication, type:

```
net start csauth
```

and press **Enter**.

---

## User and AAA Client Import Option

The `-i` option enables you to update Cisco Secure ACS with data from a colon-delimited text file. You can also update AAA client definitions.

For user accounts, you can add users, change user information such as passwords, or delete users. For AAA client definitions, you can add or delete AAA clients.

This section contains the following topics:

- [Importing User and AAA Client Information, page D-15](#)
- [User and AAA Client Import File Format, page D-16](#)
  - [About User and AAA Client Import File Format, page D-17](#)
  - [ONLINE or OFFLINE Statement, page D-17](#)
  - [ADD Statements, page D-18](#)
  - [UPDATE Statements, page D-19](#)
  - [DELETE Statements, page D-21](#)
  - [ADD\\_NAS Statements, page D-21](#)
  - [DEL\\_NAS Statements, page D-23](#)
  - [Import File Example, page D-24](#)

## Importing User and AAA Client Information

To import user or AAA client information, follow these steps:

- 
- Step 1** If you have not performed a backup or dump of Cisco Secure ACS, do so now before proceeding. For more information about backing up the database, see [Backing Up Cisco Secure ACS with CSUtil.exe](#), page D-6.
- Step 2** Create an import text file. For more information about what an import text file can or must contain, see [User and AAA Client Import File Format](#), page D-16.
- Step 3** Copy or move the import text file to the same directory as CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files](#), page D-2.
- Step 4** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe.
- Step 5** Type:
- ```
CSUtil.exe -i filename
```
- where *filename* is the name of the import text file you want CSUtil.exe to use to update Cisco Secure ACS. Press **Enter**.
- CSUtil.exe displays a confirmation prompt for updating the database.
- Step 6** To confirm that you want to update Cisco Secure ACS with the information from the import text file specified, type **Y** and press **Enter**.
- Cisco Secure ACS is updated with the information in the import text file specified. This process may take a few minutes.
- If the import text file contained AAA client configuration data, CSUtil.exe warns you that you need to restart CSTacacs and CSRADIUS for these changes to take effect.
- Step 7** To restart CSRADIUS, follow these steps:
- Type:

```
net stop csradius
```

and press **Enter**.

The CSRADIUS service stops.

- b. To start CSRadius, type:

```
net start csradius
```

and press **Enter**.

**Step 8** To restart CSTacacs, follow these steps:

- a. Type:

```
net stop cstacacs
```

and press **Enter**.

The CSTacacs service stops.

- b. To start CSTacacs, type:

```
net start cstacacs
```

and press **Enter**.

---

## User and AAA Client Import File Format

This section contains the following topics:

- [About User and AAA Client Import File Format, page D-17](#)
- [ONLINE or OFFLINE Statement, page D-17](#)
- [ADD Statements, page D-18](#)
- [UPDATE Statements, page D-19](#)
- [DELETE Statements, page D-21](#)
- [ADD\\_NAS Statements, page D-21](#)
- [DEL\\_NAS Statements, page D-23](#)
- [Import File Example, page D-24](#)

## About User and AAA Client Import File Format

The import file can contain six different line types, as discussed in following topics. The first line of the import file must be one of the tokens defined in [Table D-1](#).

Each line of a CSUtil.exe import file is a series of colon-separated tokens. Some of the tokens are followed by values. Values, like tokens, are colon-delimited. For tokens that require values, CSUtil.exe expects the value of the token to be in the colon-delimited field immediately following the token.

### ONLINE or OFFLINE Statement

CSUtil.exe requires an ONLINE or OFFLINE token in an import text file. The file must begin with a line that contains only an ONLINE or OFFLINE token. The ONLINE and OFFLINE tokens are described in [Table D-1](#).

**Table D-1** *ONLINE/OFFLINE Statement Tokens*

Token	Required	Value Required	Description
ONLINE	Either ONLINE or OFFLINE must be present	—	The CSAuth service remains active while CSUtil.exe imports the text file. CSUtil.exe performance is slower when run in this mode, but Cisco Secure ACS continues to authenticate users during the import.
OFFLINE	Either ONLINE or OFFLINE must be present	—	The CSAuth service is stopped while CSUtil.exe imports the text file. Although CSUtil.exe performance is fastest in this mode, no users are authenticated during the import.  If you need to import a large amount of user information quickly, consider using the OFFLINE token. While performing an import in the OFFLINE mode stops authentication during the import, the import is much faster. For example, importing 100,000 users in the OFFLINE mode takes less than one minute.

## ADD Statements

ADD statements are optional. Only the ADD token and its value are required to add a user to Cisco Secure ACS. The valid tokens for ADD statements are listed in [Table D-2](#).



### Note

CSUtil.exe provides no means to specify a particular instance of an external user database type. If a user is to be authenticated by an external user database and Cisco Secure ACS has multiple instances of the specified database type, CSUtil.exe assigns the user to the first instance of that database type. For example, if Cisco Secure ACS has two LDAP external user databases configured, CSUtil.exe creates the user record and assigns the user to the LDAP database that was added to Cisco Secure ACS first.

**Table D-2** ADD Statement Tokens

Token	Required	Value Required	Description
ADD	Yes	username	Add user information to Cisco Secure ACS. If the username already exists, no information is changed.
PROFILE	No	group number	Group number to which the user is assigned. This must be a number from 0 to 499, not a name. If you do not use the PROFILE token or fail to provide a group number, the user is added to the default group.
CHAP	No	CHAP password	Require a CHAP password for authentication.
CSDB	No	password	Authenticate the username with the CiscoSecure user database.
CSDB_UNIX	No	UNIX-encrypted password	Authenticate the username with the CiscoSecure user database, using a UNIX password format.
EXT_NT	No	—	Authenticate the username with a Windows external user database.
EXT_NDS	No	—	Authenticate the username with a Novell NDS external user database.

Table D-2 ADD Statement Tokens (continued)

Token	Required	Value Required	Description
EXT_SDI	No	—	Authenticate the username with an RSA external user database.
EXT_ODBC	No	—	Authenticate the username with an ODBC external user database.
EXT_LDAP	No	—	Authenticate the username with a generic LDAP external user database.
EXT_LEAP	No	—	Authenticate the username with a LEAP proxy RADIUS server external user database.
EXT_RADIUS	No	—	Authenticate the username with a RADIUS token server external user database.

For example, the following ADD statement would create an account with the username “John”, assign it to Group 3, and specify that John should be authenticated by the CiscoSecure user database with the password “closedmondays”:

```
ADD:John:PROFILE:3:CSDB:closedmondays
```

## UPDATE Statements

UPDATE statements are optional. They make changes to existing user accounts. Only the UPDATE token and its value are required by CSUtil.exe, but if no other tokens are included, no changes are made to the user account. You can use the UPDATE statement to update the group a user is assigned to or to update which database Cisco Secure ACS uses to authenticate the user. The valid tokens for UPDATE statements are listed in [Table D-3](#).

Table D-3 UPDATE Statement Tokens

Token	Required	Value Required	Description
UPDATE	Yes	username	Update user information to Cisco Secure ACS.
PROFILE	No	group number	Group number to which the user is assigned. This must be a number from 0 to 499, not a name.  <b>Note</b> If you do not specify a database token, such as CSDB or EXT_NT, updating a group assignment may erase a user's password.
CHAP	No	CHAP password	Require a CHAP password for authentication.
CSDB	No	password	Authenticate the username with the CiscoSecure user database.
CSDB_UNIX	No	UNIX-encrypted password	Authenticate the username with the CiscoSecure user database, using a UNIX password format.
EXT_NT	No	—	Authenticate the username with a Windows external user database.
EXT_NDS	No	—	Authenticate the username with a Novell NDS external user database.
EXT_ODBC	No	—	Authenticate the username with an ODBC external user database.
EXT_LDAP	No	—	Authenticate the username with a generic LDAP external user database.
EXT_LEAP	No	—	Authenticate the username with a LEAP proxy RADIUS server external user database.
EXT_RADIUS	No	—	Authenticate the username with a RADIUS token server external user database.

For example, the following UPDATE statement causes CSUtil.exe to update the account with username “John”, assign it to Group 50, specify that John should be authenticated by a UNIX-encrypted password, with a separate CHAP password “goodoldchap”:

```
UPDATE:John:PROFILE:50:CSDB_UNIX:3A13qf9:CHAP:goodoldchap
```

## DELETE Statements

DELETE statements are optional. The DELETE token and its value are required to delete a user account from Cisco Secure ACS. The DELETE token, detailed in [Table D-4](#), is the only token in a DELETE statement.

*Table D-4 UPDATE Statement Tokens*

Token	Required	Value Required	Description
DELETE	Yes	username	The name of the user account that is to be deleted.

For example, the following DELETE statement causes CSUtil.exe to permanently remove the account with username “John” from the CiscoSecure user database:

```
DELETE:John
```

## ADD\_NAS Statements

ADD\_NAS statements are optional. The ADD\_NAS, IP, KEY, and VENDOR tokens and their values are required to add a AAA client definition to Cisco Secure ACS. The valid tokens for ADD\_NAS statements are listed in [Table D-5](#).

Table D-5 ADD\_NAS Statement Tokens

Token	Required	Value Required	Description
ADD_NAS	Yes	AAA client name	The name of the AAA client that is to be added.
IP	Yes	IP address	The IP address of the AAA client being added. Use a pipe ( ) between IP addresses to import devices with multiple IPs.
KEY	Yes	Shared secret	The shared secret for the AAA client.
VENDOR	Yes	See description	<p>The authentication protocol the AAA client uses. For RADIUS, this includes the VSA.</p> <p><b>Note</b> The valid values are listed below. Quotation marks are required due to the spaces in the protocol names.</p> <ul style="list-style-type: none"> <li>• “TACACS+ (Cisco IOS)”</li> <li>• “RADIUS (Cisco Aironet)”</li> <li>• “RADIUS (Cisco BBSM)”</li> <li>• “RADIUS (Cisco IOS/PIX)”</li> <li>• “RADIUS (Cisco VPN 3000)”</li> <li>• “RADIUS (Cisco VPN 5000)”</li> <li>• “RADIUS (IETF)”</li> <li>• “RADIUS (Ascend)”</li> <li>• “RADIUS (Juniper)”</li> <li>• “RADIUS (Nortel)”</li> <li>• “RADIUS (iPass)”</li> </ul>
NDG	No	NDG name	The name of the Network Device Group to which the AAA client is to be added.

Table D-5 ADD\_NAS Statement Tokens (continued)

Token	Required	Value Required	Description
SINGLE_CON	No	Y or N	For AAA clients using TACACS+ only, the value set for this TOKEN specifies whether the Single Connect TACACS+ AAA Client option is enabled. For more information, see <a href="#">Adding a AAA Client, page 4-16</a> .
KEEPALIVE	No	Y or N	For AAA clients using TACACS+ only, the value set for this token specifies whether the Log Update/Watchdog Packets from this Access Server option is enabled. For more information, see <a href="#">Adding a AAA Client, page 4-16</a> .

For example, the following ADD\_NAS statement causes CSUtil.exe to add a AAA client with the name “SVR2-T+”, using TACACS+ with the single connection and keep alive packet options enabled:

```
ADD_NAS:SVR2-T+:IP:IP address:KEY:shared secret:VENDOR:"TACACS+ (Cisco IOS)":NDG:"East Coast":SINGLE_CON:Y:KEEPALIVE:Y
```

## DEL\_NAS Statements

DEL\_NAS statements are optional. The DEL\_NAS token, detailed in [Table D-6](#), is the only token in a DEL\_NAS statement. DEL\_NAS statements delete AAA client definitions from Cisco Secure ACS.

Table D-6 DEL\_NAS Statement Tokens

Token	Required	Value Required	Description
DEL_NAS	Yes	AAA client name	The name of the AAA client that is to be deleted.

For example, the following DEL\_NAS statement causes CSUtil.exe to delete a AAA client with the name “SVR2-T+”:

```
DEL_NAS:SVR2-T+
```

## Import File Example

The following is an example import text file:

```
OFFLINE
ADD:user01:CSDB:userpassword:PROFILE:1
ADD:user02:EXT_NT:PROFILE:2
ADD:chapuser:CSDB:hello:CHAP:chappw:PROFILE:3
ADD:mary:EXT_NT:CHAP:achappassword
ADD:joe:EXT_SDI
ADD:vanessa:CSDB:vanessapassword
ADD:juan:CSDB_UNIX:unixpassword
UPDATE:foobar:PROFILE:10
DELETE:paul
ADD_NAS:SVR2-T+:IP:209.165.202.136:KEY:A87il032bzg:VENDOR:"TACACS+ (Cisco IOS)":NDG:"East Coast"
DEL_NAS:SVR16-RAD
```

## Exporting User List to a Text File

You can use the `-u` option to export a list of all users in the CiscoSecure user database to a text file named `users.txt`. The `users.txt` file organizes users by group. Within each group, users are listed in the order that their user accounts were created in the CiscoSecure user database. For example, if accounts were created for Pat, Dana, and Lloyd, in that order, `users.txt` lists them in that order as well, rather than alphabetically.



### Note

Using the `-u` option requires that you stop the CSAAuth service. While CSAAuth is stopped, no users are authenticated.

To export user information from the CiscoSecure user database into a text file, follow these steps:

- Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** If the CSAuth service is running, type:

```
net stop csauth
```

and press **Enter**.

The CSAuth service stops.

**Step 3** Type:

```
CSUtil.exe -u
```

and press **Enter**.

CSUtil.exe exports information for all users in the CiscoSecure user database to a file named `users.txt`.

**Step 4** To resume user authentication, type:

```
net start csauth
```

and press **Enter**.

---

## Exporting Group Information to a Text File

You can use the `-g` option to export group configuration data, including device command sets, from the CiscoSecure user database to a text file named `groups.txt`. The `groups.txt` file is useful primarily for debugging purposes while working with the TAC.



### Note

Using the `-g` option requires that you stop the CSAuth service. While CSAuth is stopped, no users are authenticated.

---

To export group information from the CiscoSecure user database to a text file, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

## Exporting Registry Information to a Text File

**Step 2** If the CSAuth service is running, type:

```
net stop csauth
```

and press **Enter**.

The CSAuth service stops.

**Step 3** Type:

```
CSUtil.exe -g
```

and press **Enter**.

CSUtil.exe exports information for all groups in the CiscoSecure user database to a file named `groups.txt`.

**Step 4** To resume user authentication, type:

```
net start csauth
```

and press **Enter**.

---

# Exporting Registry Information to a Text File

You can use the `-y` option to export Windows Registry information for Cisco Secure ACS. CSUtil.exe exports the Registry information to a file named `setup.txt`. The `setup.txt` file is primarily useful for debugging purposes while working with the TAC.

To export Registry information from Cisco Secure ACS to a text file, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** Type:

```
CSUtil.exe -y
```

and press **Enter**.

CSUtil.exe exports Windows Registry information for Cisco Secure ACS to a file named `setup.txt`.

---

## Decoding Error Numbers

You can use the `-e` option to decode error numbers found in Cisco Secure ACS service logs. These are error codes internal to Cisco Secure ACS. For example, the CSRADIUS log could contain a message similar to the following:

```
CSRADIUS/Logs/RDS.log:RDS 05/22/2001 10:09:02 E 2152 4756 Error -1087 authenticating geddy  
- no NAS response sent
```

In this example, the error code number that you could use CSUtil.exe to decode is “-1087”:

```
C:\Program Files\CiscoSecure ACS vX.X\Utils: CSUtil.exe -e -1087  
CSUtil v3.0(1.14), Copyright 1997-2001, Cisco Systems Inc  
Code -1087 : External database reported error during authentication
```



### Note

The `-e` option applies to Cisco Secure ACS internal error codes only, not to Windows error codes sometimes captured in Cisco Secure ACS logs, such as when Windows authentication fails.

---

For more information about Cisco Secure ACS service logs, see [Service Logs, page 11-31](#).

To decode an error number from a Cisco Secure ACS service log, follow these steps:

- 
- Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

Step 2 Type:

```
CSUtil.exe -e -number
```

where *number* is the error number found in the Cisco Secure ACS service log. Press **Enter**.



---

**Note** The hyphen (-) before *number* is required.

---

CSUtil.exe displays the text message equivalent to the error number specified.

---

## Recalculating CRC Values

The -c option is for use by the TAC. Its purpose is to resolve CRC (cyclical redundancy check) value conflicts between files manually copied into your Cisco Secure ACS directories and the values recorded in the Windows Registry.



---

**Note** Do not use the -c option unless a Cisco representative requests that you do.

---

## User-Defined RADIUS Vendors and VSA Sets

This section provides information and procedures about user-defined RADIUS vendors and VSAs.

This section contains the following topics:

- [About User-Defined RADIUS Vendors and VSA Sets, page D-29](#)
- [Adding a Custom RADIUS Vendor and VSA Set, page D-29](#)
- [Deleting a Custom RADIUS Vendor and VSA Set, page D-31](#)
- [Listing Custom RADIUS Vendors, page D-32](#)
- [Exporting Custom RADIUS Vendor and VSA Sets, page D-33](#)
- [RADIUS Vendor/VSA Import File, page D-34](#)

## About User-Defined RADIUS Vendors and VSA Sets

In addition to supporting a set of predefined RADIUS vendors and vendor-specific attributes (VSAs), Cisco Secure ACS supports RADIUS vendors and VSAs that you define. We recommend that you use RDBMS Synchronization to add and configure custom RADIUS vendors; however, you can use CSUtil.exe to accomplish the same custom RADIUS vendor and VSA configurations that you can accomplish using RDBMS Synchronization. Custom RADIUS vendor and VSA configuration created by either of these two features—RDBMS Synchronization or CSUtil.exe—can be modified by the other feature. Choosing one feature for configuring custom RADIUS vendors and VSAs does not preclude using the other feature. For more information about RDBMS Synchronization, see [RDBMS Synchronization, page 9-25](#).

Vendors you add must be IETF-compliant; therefore, all VSAs that you add must be sub-attributes of IETF RADIUS attribute number 26. You can define up to ten custom RADIUS vendors, numbered 0 (zero) through 9. CSUtil.exe allows only one instance of any given vendor, as defined by the unique vendor IETF ID number and by the vendor name.



---

**Note**

If you intend to replicate user-defined RADIUS vendor and VSA configurations, user-defined RADIUS vendor and VSA definitions to be replicated must be identical on the primary and secondary Cisco Secure ACSes, including the RADIUS vendor slots that the user-defined RADIUS vendors occupy. For more information about database replication, see [CiscoSecure Database Replication, page 9-1](#).

---

## Adding a Custom RADIUS Vendor and VSA Set

You can use the `-addUDV` option to add up to ten custom RADIUS vendors and VSA sets to Cisco Secure ACS. Each RADIUS vendor and VSA set is added to one of ten possible user-defined RADIUS vendor slots.



---

**Note**

While CSUtil.exe adds a custom RADIUS vendor and VSA set to Cisco Secure ACS, all Cisco Secure ACS services are automatically stopped and restarted. No users are authenticated during this process.

---

### Before You Begin

- Define a custom RADIUS vendor and VSA set in a RADIUS vendor/VSA import file. For more information, see [RADIUS Vendor/VSA Import File, page D-34](#).
- Determine the RADIUS vendor slot to which you want to add the new RADIUS vendor and VSAs. For more information, see [Listing Custom RADIUS Vendors, page D-32](#).
- Make sure that regedit is not running. If regedit is running on the Cisco Secure ACS Windows server, it can prevent Registry updates required for adding a custom RADIUS vendor and VSA set.

To add a custom RADIUS VSA to Cisco Secure ACS, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** Type:

```
CSUtil.exe -addUDV slot-number
filename
```

where *slot-number* is an unused Cisco Secure ACS RADIUS vendor slot and *filename* is the name of a RADIUS vendor/VSA import file. The *filename* can include a relative or absolute path to the RADIUS vendor/VSA import file. Press **Enter**.

For example, to add the RADIUS vendor defined in d:\acs\myvsa.ini to slot 5, the command would be:

```
CSUtil.exe -addUDV 5 d:\acs\myvsa.ini
```

CSUtil.exe displays a confirmation prompt.

**Step 3** To confirm that you want to add the RADIUS vendor and halt all Cisco Secure ACS services during the process, type **Y** and press **Enter**.

CSUtil.exe halts Cisco Secure ACS services, parses the vendor/VSA input file, and adds the new RADIUS vendor and VSAs to Cisco Secure ACS. This process may take a few minutes. After it is complete, CSUtil.exe restarts Cisco Secure ACS services.



---

**Note** We recommend that you archive RADIUS vendor/VSA import files. During upgrades, the Utils directory, where CSUtil.exe is located, is replaced, including all its contents. Backing up RADIUS vendor/VSA import files ensures that you can recover your custom RADIUS vendors and VSAs after reinstallation or upgrading to a later release.

---

## Deleting a Custom RADIUS Vendor and VSA Set

You can use the `-delUDV` option to delete a custom RADIUS vendor from Cisco Secure ACS.



---

**Note** While CSUtil.exe deletes a custom RADIUS vendor from Cisco Secure ACS, all Cisco Secure ACS services are automatically stopped and restarted. No users are authenticated while this process is occurring.

---

### Before You Begin

Verify that, in the Network Configuration section of the Cisco Secure ACS HTML interface, no AAA client uses the RADIUS vendor. For more information about configuring AAA clients, see [AAA Client Configuration, page 4-11](#).

Verify that your RADIUS accounting log does not contain attributes from the RADIUS vendor you want to delete. For more information about configuring your RADIUS accounting log, see [Accounting Logs, page 11-6](#).

To delete a custom RADIUS vendor and VSA set from Cisco Secure ACS, follow these steps:

- 
- Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

**Step 2** Type:

```
CSUtil.exe -delUDV slot-number
```

where *slot-number* is the slot containing the RADIUS vendor that you want to delete. Press **Enter**.




---

**Note** For more information about determining what RADIUS vendor a particular slot contains, see [Listing Custom RADIUS Vendors, page D-32](#).

---

CSUtil.exe displays a confirmation prompt.

**Step 3** To confirm that you want to halt all Cisco Secure ACS services while deleting the custom RADIUS vendor and VSAs, type **Y** and press **Enter**.

CSUtil.exe displays a second confirmation prompt.

**Step 4** To confirm that you want to delete the RADIUS vendor, type **Y** and press **Enter**.

CSUtil.exe halts Cisco Secure ACS services, deletes the specified RADIUS vendor from Cisco Secure ACS. This process may take a few minutes. After it is complete, CSUtil.exe restarts Cisco Secure ACS services.

---

## Listing Custom RADIUS Vendors

You can use the `-listUDV` option to determine what custom RADIUS vendors are defined in Cisco Secure ACS. This option also enables you to determine which of the ten possible custom RADIUS vendor slots are in use and which RADIUS vendor occupies each used slot.

To list all custom RADIUS vendors defined in Cisco Secure ACS, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

Step 2 Type:

```
CSUtil.exe -listUDV
```

Press **Enter**.

CSUtil.exe lists each user-defined RADIUS vendor slot in slot number order. CSUtil.exe lists slots that do not contain a custom RADIUS vendor as “Unassigned”. An unassigned slot is empty. You can add a custom RADIUS vendor to any slot listed as “Unassigned”.

---

## Exporting Custom RADIUS Vendor and VSA Sets

You can export all custom RADIUS vendor and VSA sets to files. Each vendor and VSA set is saved to a separate file. The files created by this option are in the same format as RADIUS vendor/VSA import files. This option is particularly useful if you need to modify a custom RADIUS vendor and VSA set and you have misplaced the original file used to import the set.



Note

Exporting a custom RADIUS vendor and VSA set does not remove the vendor and VSA set from Cisco Secure ACS.

---

Cisco Secure ACS places all exported vendor/VSA files in a subdirectory of the directory containing CSUtil.exe. The subdirectory is named `system UDV`s. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

Each exported vendor/VSA file is named `UDV_n.ini`, where *n* is the slot number currently occupied by the custom RADIUS vendor and VSA set. For example, if vendor Widget occupies slot 4, the exported file created by CSUtil.exe is `UDV_4.ini`.

To export custom RADIUS vendor and VSA sets to files, follow these steps:

---

Step 1 On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe. For more information about the location of CSUtil.exe, see [Location of CSUtil.exe and Related Files, page D-2](#).

Step 2 Type:

```
CSUtil.exe -dumpUDV
```

Press **Enter**.

For each custom RADIUS vendor and VSA set currently configured in Cisco Secure ACS, CSUtil.exe writes a file in the `system UDV`s subdirectory.

---

## RADIUS Vendor/VSA Import File

To import a custom RADIUS vendor and VSA set into Cisco Secure ACS, you must define the RADIUS vendor and VSA set in an import file. This section details the format and content of RADIUS VSA import files.

We recommend that you archive RADIUS vendor/VSA import files. During upgrades, the `utils` directory, where CSUtil.exe is located, is replaced, including all its contents. Backing up RADIUS vendor/VSA import files ensures that you can recover your custom RADIUS vendors and VSAs after reinstallation or upgrading to a later release.

This section contains the following topics:

- [About the RADIUS Vendor/VSA Import File, page D-34](#)
- [Vendor and VSA Set Definition, page D-35](#)
- [Attribute Definition, page D-36](#)
- [Enumeration Definition, page D-38](#)
- [Example RADIUS Vendor/VSA Import File, page D-39](#)

## About the RADIUS Vendor/VSA Import File

RADIUS Vendor/VSA import files use a Windows .ini file format. Each RADIUS vendor/VSA import file comprises three types of sections, detailed in [Table D-7](#). Each section comprises a section header and a set of keys and values. The order of the sections in the RADIUS vendor/VSA import file is irrelevant.

**Table D-7 RADIUS VSA Import File Section Types**

Section	Required	Number	Description
Vendor and VSA set definition	Yes	1	Defines the RADIUS vendor and VSA set. For more information, see <a href="#">Vendor and VSA Set Definition, page D-35</a> .
Attribute definition	Yes	1 to 255	Defines a single attribute of the VSA set. For more information, see <a href="#">Attribute Definition, page D-36</a> .
Enumeration	No	0 to 255	Defines enumerations for attributes with integer data types. For more information, see <a href="#">Enumeration Definition, page D-38</a> .

## Vendor and VSA Set Definition

Each RADIUS vendor/VSA import file must have one vendor and VSA set section. The section header must be “[User Defined Vendor]”. [Table D-8](#) lists valid keys for the vendor and VSA set section.

**Table D-8 Vendor and VSA Set Keys**

Keys	Required	Value Required	Description
Name	Yes	Vendor name	The name of the RADIUS vendor.
IETF Code	Yes	An integer	The IETF-assigned vendor number for this vendor.
VSA <i>n</i> (where <i>n</i> is the VSA number)	Yes—you can define 1 to 255 VSAs	Attribute name	<p>The name of a VSA. For each VSA named here, the file must contain a corresponding attribute definition section.</p> <p><b>Note</b> Attribute names must be unique within the RADIUS vendor/VSA import file, and within the set of all RADIUS attributes in Cisco Secure ACS. To facilitate this, we recommend that you prefix the vendor name to each attribute name, such as “widget-encryption” for an encryption-related attribute for the vendor Widget. This also makes accounting logs easier to understand.</p>

For example, the following vendor and VSA set section defines the vendor “Widget”, whose IETF-assigned vendor number is 9999. Vendor Widget has 4 VSAs (thus requiring 4 attribute definition sections):

```
[User Defined Vendor]
Name=Widget
IETF Code=9999
VSA 1=widget-encryption
VSA 2=widget-admin-interface
VSA 3=widget-group
VSA 4=widget-admin-encryption
```

## Attribute Definition

Each RADIUS vendor/VSA import file must have one attribute definition section for each attribute defined in the vendor and VSA set section. The section header of each attribute definition section must match the attribute name defined for that attribute in the vendor and VSA set section. [Table D-8](#) lists the valid keys for an attribute definition section.

Table D-9 Attribute Definition Keys

Keys	Required	Value Required	Description
Type	Yes	See Description	<p>The data type of the attribute. It must be one of the following:</p> <ul style="list-style-type: none"> <li>• STRING</li> <li>• INTEGER</li> <li>• IPADDR</li> </ul> <p>If the attribute is an integer, the Enums key is valid.</p>
Profile	Yes	See Description	<p>The attribute profile defines if the attribute is used for authorization or accounting (or both). At least one of the following two values must be present in the Profile key definition:</p> <ul style="list-style-type: none"> <li>• <b>IN</b>—The attribute is used for accounting. After you add the attribute to Cisco Secure ACS, you can configure your RADIUS accounting log to record the new attribute. For more information about RADIUS accounting logs, see <a href="#">Accounting Logs, page 11-6</a>.</li> <li>• <b>OUT</b>—The attribute is used for authorization.</li> </ul> <p>In addition, you can use the value “MULTI” to allow several instances of the attribute per RADIUS message. Combinations are valid. For example:</p> <pre>Profile=MULTI OUT</pre> <p>or</p> <pre>Profile=IN OUT</pre>
Enums	No (only valid when the TYPE value is INTEGER)	Enumerations section name	<p>The name of the enumeration section.</p> <p><b>Note</b> Several attributes can reference the same enumeration section. For more information, see <a href="#">Enumeration Definition, page D-38</a>.</p>

For example, the following attribute definition section defines the widget-encryption VSA, which is an integer used for authorization, and for which enumerations exist in the Encryption-Types enumeration section:

```
[widget-encryption]  
Type=INTEGER  
Profile=OUT  
Enums=Encryption-Types
```

## Enumeration Definition

Enumeration definitions enable you to associate a text-based name for each valid numeric value of an integer-type attribute. In the Group Setup and User Setup sections of the Cisco Secure ACS HTML interface, the text values you define appear in lists associated with the attributes that use the enumerations. Enumeration definition sections are required only if an attribute definition section references them. Only attributes that are integer-type attributes can reference an enumeration definition section.

The section header of each enumeration definition section must match the value of an Enums key that references it. An enumeration definition section can be referenced by more than one Enums key, thus allowing for reuse of common enumeration definitions. An enumeration definition section can have up to 1000 keys.

[Table D-10](#) lists the valid keys for an enumeration definition section.

Table D-10 Enumerations Definition Keys

Keys	Required	Value Required	Description
<i>n</i> (See description.)	Yes	String	<p>For each valid integer value of the corresponding attribute, an enumerations section must have one key.</p> <p>Each key defines a string value associated with an integer value. Cisco Secure ACS uses these string values in the HTML interface.</p> <p>For example, if 0 through 4 are valid integer values for a given attribute, its enumeration definition would contain the following:</p> <pre>0=value0 1=value1 2=value2 3=value3 4=value4</pre>

For example, the following enumerations definition section defines the Encryption-Types enumeration, which associates the string value 56-bit with the integer 0 and the string value 128-bit with the integer 1:

```
[Encryption-Types]
0=56-bit
1=128-bit
```

## Example RADIUS Vendor/VSA Import File

The example RADIUS vendor/VSA import file, below, defines the vendor Widget, whose IETF number is 9999. The vendor Widget has 5 VSAs. Of those attributes, 4 are for authorization and one is for accounting. Only one attribute can have multiple instances in a single RADIUS message. Two attributes have enumerations for their valid integer values and they share the same enumeration definition section.

```
[User Defined Vendor]
Name=Widget
IETF Code=9999
VSA 1=widget-encryption
```

```
VSA 2=widget-admin-interface
VSA 3=widget-group
VSA 4=widget-admin-encryption
VSA 5=widget-remote-address
```

```
[widget-encryption]
Type=INTEGER
Profile=OUT
Enums=Encryption-Types
```

```
[widget-admin-interface]
Type=IPADDR
Profile=OUT
```

```
[widget-group]
Type=STRING
Profile=MULTI OUT
```

```
[widget-admin-encryption]
Type=INTEGER
Profile=OUT
Enums=Encryption-Types
```

```
[widget-remote-address]
Type=STRING
Profile=IN
```

```
[Encryption-Types]
0=56-bit
1=128-bit
2=256-bit
```

## PAC File Generation

You can use the `-t` option to generate PAC files for use with EAP-FAST clients. For more information about PACs and EAP-FAST, see [EAP-FAST Authentication, page 10-13](#).

This section contains the following topics:

- [PAC File Options and Examples, page D-41](#)
- [Generating PAC Files, page D-43](#)

## PAC File Options and Examples

When you use the `-t` option generate PAC files with CSUtil.exe, you have the following additional options.

- **User specification options**—While you can choose which user specification option you want to use, you **must** choose one of the four options for specifying which users you want PAC files for; otherwise, CSUtil.exe displays an error message because no users are specified. User specification options are as follows:
  - **-a**—CSUtil.exe generates a PAC file for each user in the CiscoSecure user database. For example, if you have 3278 users in the CiscoSecure user database and ran **CSUtil.exe -t -a**, CSUtil.exe would generate 3278 PAC files, one for each user.



---

**Note** Using the `-a` option restarts the CSAuth service. No users are authenticated while CSAuth is unavailable.

---

- **-g N**—CSUtil.exe generates a PAC file for each user in the user group specified by number (*N*). Cisco Secure ACS has 500 groups, numbered from 0 (zero) to 499. For example, if group 7 has 43 users and you ran **CSUtil.exe -t -g 7**, CSUtil.exe would generate 43 PAC files, one for each user who is a member of group 7.



---

**Note** Using the `-g` option restarts the CSAuth service. No users are authenticated while CSAuth is unavailable.

---

- **-u username**—CSUtil.exe generates a PAC file for the user specified by name (*username*). For example, if you ran **CSUtil.exe -t -u seaniemop**, CSUtil.exe would generate a single PAC file, named `seaniemop.pac`.



**Tip**

---

You can also specify a domain-qualified username, using the format `DOMAIN\username`. For example, if you specify `ENGINEERING\augustin`, Cisco Secure ACS generates a PAC file name `ENGINEERING_augustin.pac`.

---

- **-f list**—CSUtil.exe generates a PAC file for each username contained in the file specified, where *list* represents the full path and filename of the list of usernames.

Lists of usernames should contain one username per line with no additional spaces or other characters.

For example, if list.txt in d:\temp\pacs contains the following usernames:

```
seaniemop
jwiedman
echamberlain
```

and you ran **CSUtil.exe -t -f d:\temp\pacs\list.txt**, CSUtil.exe generates three PAC files: seaniemop.pac, jwiedman.pac, and echamberlain.pac.



#### Tip

You can also specify domain-qualified usernames, using the format *DOMAIN\username*. For example, if you specify `ENGINEERING\augustin`, Cisco Secure ACS generates a PAC file name `ENGINEERING_augustin.pac`.

- **-passwd password**—CSUtil.exe uses the password specified, rather than the default password, to protect the PAC files it generates. The password you specify is required when the PACs it protects are loaded into an EAP-FAST end-user client.



#### Note

We recommend that you use a password you devise rather than the default password.

PAC passwords can contain any character, are between four and 128 characters long, and case sensitive. While CSUtil.exe does not enforce strong password rules, we recommend that you use a strong password, that is, your PAC password should:

- Be very long.
- Contain uppercase and lowercase letters.
- Contain numbers in addition to letters.
- Contain no common words or names.

## Generating PAC Files

**Note**

If you use the `-a` or `-g` option during PAC file generation, CSUtil.exe restarts the CSAuth service. No users are authenticated while CSAuth is unavailable.

For more information about PACs, see [About PACs, page 10-17](#).

To generate PAC files, follow these steps:

- Step 1** Use the discussion in [PAC File Options and Examples, page D-41](#), to determine the following:
- Which users you want to generate PAC files for. If you want to use a list of users, create it now.
  - What password you want to use to protect the PAC files you generate. If necessary, create a password. We recommend passwords that are long, use uppercase and lowercase letters, and include numbers.
  - The full path to the directory you want the PAC files to be created in. If necessary, create the directory.
- Step 2** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe.

**Step 3** Type

```
CSUtil.exe -t additional arguments
```

where *additional arguments* represents at least one option for specifying which users to generate PAC files for. You can also use the options to specify filepath and password.

Press **Enter**.

CSUtil.exe generates the PAC files for each user specified. The PAC files are named with the username plus a “.pac” file extension. For example, a PAC file for the username `seaniemop` would be `seaniemop.pac` and a PAC file for the domain-qualified username `ENGINEERING\augustin` would be `ENGINEERING_augustin.pac`.

If you specified a filepath, the PAC files are saved where you specified. You can distribute the PAC files to the applicable end-user clients.

---

## Posture Validation Attributes

You can use CSUtil.exe to export, add, and delete posture validation attributes, which are essential to Network Admission Control (NAC). For more information about NAC, see [Chapter 14, “Network Admission Control”](#).

This section contains the following topics:

- [Posture Validation Attribute Definition File, page D-44](#)
- [Exporting Posture Validation Attribute Definitions, page D-48](#)
- [Importing Posture Validation Attribute Definitions, page D-49](#)
- [Deleting a Posture Validation Attribute Definition, page D-51](#)
- [Default Posture Validation Attribute Definition File, page D-52](#)

## Posture Validation Attribute Definition File

A posture validation attribute definition file is a text file that contains one or more posture validation attribute definitions. Each definition consists of a definition header and several values, described below. For an example of the contents of a posture validation attribute definition file, see [Default Posture Validation Attribute Definition File, page D-52](#).

With the exception of the attribute definition header, each attribute definition value must be formatted as follows:

*name=value*

where *name* is the value name and *value* is a string or integer, as specified in the list below.



**Tip**

---

Use a semi-colon to identify lines that are comments.

---

[Example D-1](#) shows an example of a posture validation attribute definition, including a comment after the attribute definition:

### **Example D-1 Example Attribute Definition**

```
[attr#0]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00001
attribute-name=Application-Posture-Token
attribute-profile=out
attribute-type=unsigned integer

; attribute 1 is reserved for the APT
```

A posture validation attribute is uniquely defined by the combination of its vendor ID, application ID, and attribute ID. The following list provides details of these values and of each line required in an attribute definition:

- **[attr#*n*]**—Attribute definition header, where *n* is a unique, sequential integer, beginning with zero. CSUtil.exe uses the definition header to distinguish the beginning of a new attribute definition. Each attribute definition *must* begin with a line containing the definition header. The first attribute definition in the file *must* have the header [attr#0], the second attribute definition in a file must have the header [attr#1], and so on. A break in the numbering causes CSUtil.exe to ignore attribute definitions at the break and beyond. For example, if a file with 10 attribute definitions the fifth attribute is defined as [attr#5] instead of [attr#4], CSUtil.exe ignores the attribute defined as [attr#5] and remaining five the attributes following it.



#### **Tip**

---

The value of *n* is irrelevant to any of the ID values in the attribute definition file. For example, the 28th definition in a file must have the header [attr#27], but this does not limit or otherwise define valid values for vendor-id, application-id, attribute-id. Neither does it limit or define the number of posture validation attributes supported by Cisco Secure ACS.

---

- **vendor-id**—An unsigned integer, the vendor number is of the vendor associated with the posture validation attribute. The vendor number should be the number assigned to the vendor in the [IANA Assigned Numbers RFC](#). For example, vendor ID 9 corresponds to Cisco Systems, Inc.

Vendor IDs have one or more applications associated with them, identified by the application-id value.

- **vendor-name**—A string, the vendor name appears in the Cisco Secure ACS HTML interface and logs for the associated posture validation attribute. For example, any attribute definition with a vendor ID of 9 could have a vendor name “Cisco”.




---

**Note** The vendor name cannot differ for each attribute that shares the same vendor ID. For example, you cannot add an attribute with a vendor-id of 9 if the vendor-name is not “Cisco”.

---

- **application-id**—An unsigned integer, the application ID uniquely identifies the vendor application associated with the posture validation attribute. For example, if the vendor ID is 9 and the application ID is 1, the posture validation attribute is associated with the Cisco application with an ID of 1, which is the Cisco Trust Agent (CTA), also known as a posture agent (PA).
- **application-name**—A string, the application name appears in the Cisco Secure ACS HTML interface and logs for the associated posture validation attribute. For example, if the vendor ID is 9 and the application ID is 1, the application name would be “PA”, an abbreviation of posture agent, which is another term for CTA.




---

**Note** The application name cannot differ for each attribute that shares the same vendor ID and application ID pair. For example, you cannot add an attribute with a vendor-id of 9 and application ID of 1 if the application-name is not “PA”.

---

- **attribute-id**—An unsigned integer in the range of 1 to 65535, the attribute ID uniquely identifies the posture validation attribute for the vendor ID and application ID specified.




---

**Note** For each application, attributes 1 and 2 are reserved. If you add attributes that imply a new application, CSUtil.exe automatically creates attribute 1 as Application-Posture-Token and attribute 2 as System-Posture-Token.

---

- **attribute-name**—A string, the attribute name appears in the Cisco Secure ACS HTML interface and logs for the associated posture validation attribute. For example, if the vendor ID is 9, the application ID is 1, and the attribute ID is 1, the attribute name is “Application-Posture-Token”.
- **attribute-profile**—A string, the attribute profile specifies whether Cisco Secure ACS can send the attribute in a posture validation response, can receive the attribute in a posture validation request, or can both send and receive the attribute during posture validation. Valid values for attribute-profile are:
  - **in**—Cisco Secure ACS accepts the attribute in posture validation requests and can log the attribute, and you can use it in local policy rule definitions. Attributes with an “in” attribute-profile are also known as inbound attributes.
  - **out**—Cisco Secure ACS can send the attribute in posture validation responses but you cannot use it in local policy rule definitions. Attributes with an “out” attribute-profile are also known as outbound attributes. The only outbound attributes that you can configure Cisco Secure ACS to log are the attributes for Application Posture Tokens and System Posture Tokens; however, these are system-defined attributes that you cannot modify.
  - **in out**—Cisco Secure ACS both accepts the attribute in posture validation requests and can send the attribute in posture validation responses. Attributes with an “in out” attribute-profile are also known as both inbound and outbound attributes.
- **attribute-type**—A string, the attribute type specifies the kind of data that is valid in the associated attribute. For attributes whose attribute-profile is `in` or `in out`, the attribute-type determines the types of operators available for defining local policy rules that use the attribute. An example of an inbound attribute is the ServicePacks attribute sent by CTA. An example of an outbound attribute is the System-Posture-Token attribute, sent to CTA.

Valid values of attribute-type are:

- boolean
- string
- integer
- unsigned integer
- ipaddr

- date
- version
- octet-array

For more information about attribute data types, see [NAC Attribute Data Types, page 14-19](#).

## Exporting Posture Validation Attribute Definitions

The `-dumpAVP` option exports the current posture validation attributes to an attribute definition file. For an explanation of the contents of a posture validation attribute definition file, see [Posture Validation Attribute Definition File, page D-44](#). For an example of an attribute definition file, see [Default Posture Validation Attribute Definition File, page D-52](#).

To export posture validation attributes, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe.

**Step 2** Type:

```
CSUtil.exe -dumpavp filename
```

where *filename* is the name of the file in which you want CSUtil.exe to write all attribute definitions. Press **Enter**.




---

**Tip** When you specify *filename*, you can prefix the filename with a relative or absolute path, too. For example, `CSUtil.exe -dumpavp c:\temp\allavp.txt` writes the file `allavp.txt` in `c:\temp`.

---

**Step 3** If you are prompted to confirm overwriting a file with the same path and name that you specified in [Step 2](#), do one of the following:

- To overwrite the file, type **Y** and press **Enter**.




---

**Tip** To force CSUtil.exe to overwrite an existing file, use the `-q` option:  
**CSUtil.exe -q -dumpavp *filename***.

---

- To preserve the file, type **N**, press **Enter**, and return to [Step 2](#).

CSUtil.exe writes all posture validation attribute definitions in the file specified. To view the contents of the file, use the text editor of your choice.

---

## Importing Posture Validation Attribute Definitions

The `-addAVP` option imports into Cisco Secure ACS posture validation attribute definitions from an attribute definition file. For an explanation of the contents of a posture validation attribute definition file, see [Posture Validation Attribute Definition File, page D-44](#). For an example of an attribute definition file, see [Default Posture Validation Attribute Definition File, page D-52](#).

### Before You Begin

Because completing this procedure requires restarting the CSAuth service, which temporarily suspends authentication services, consider performing this procedure when demand for Cisco Secure ACS services is low.

Use the steps in [Exporting Posture Validation Attribute Definitions, page D-48](#), to create a backup of posture validation attribute definitions. You can also use the exported attribute definition file to double-check the vendor ID, application ID, and attribute ID of current posture validation attributes.

To import posture validation attributes, follow these steps:

- 
- Step 1** Use the discussion in [Posture Validation Attribute Definition File, page D-44](#), to create a properly formatted attribute definition file. Place the file either in the directory containing CSUtil.exe or a directory accessible from the computer running Cisco Secure ACS.
  - Step 2** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe.
  - Step 3** Type:

```
CSUtil.exe -addavp filename
```

where *filename* is the name of the file in which you want CSUtil.exe to write all attribute definitions. Press **Enter**.

**Tip**

When you specify *filename*, you can prefix the filename with a relative or absolute path, too. For example, `CSUtil.exe -addavp c:\temp\addavp.txt` writes the file `addavp.txt` in `c:\temp`.

CSUtil.exe adds or modifies the attributes specified in the file. An example of a successful addition of nine posture validation attributes follows:

```
C:\...\Utils 21: csutil -addavp myavp.txt
CSUtil v3.3(1.6), Copyright 1997-2001, Cisco Systems Inc
Attribute 9876:1:11 (Calliope) added to registry
Attribute 9876:1:3 (Clio) added to registry
Attribute 9876:1:4 (Erato) added to registry
Attribute 9876:1:5 (Euterpe) added to registry
Attribute 9876:1:6 (Melpomene) added to registry
Attribute 9876:1:7 (Polyhymnia) added to registry
Attribute 9876:1:8 (Terpsichore) added to registry
Attribute 9876:1:9 (Thalia) added to registry
Attribute 9876:1:10 (Urania) added to registry
```

AVPs from 'myavp.txt' were successfully added

**Step 4**

If you are ready to make the imported attribute definitions take effect, restart the CSAuth and CSAdmin services.

**Caution**

While CSAuth is stopped, no users are authenticated.

To restart the CSAuth, CSLog, and CSAdmin services, enter the following commands at the command prompt, allowing the computer time to perform each command:

```
net stop csauth
net start csauth
net stop cslog
net start cslog
net stop csadmin
net start csadmin
```

Cisco Secure ACS begins using the imported posture validation attributes. Attributes that have an attribute type of `in` or `in out` are available in the HTML interface when you define local policy rules.

## Deleting a Posture Validation Attribute Definition

The `-delAVP` option deletes a single posture validation attribute from Cisco Secure ACS.

### Before You Begin

Because completing this procedure requires restarting the CSAuth service, which temporarily suspends authentication services, consider performing this procedure when demand for Cisco Secure ACS services is low.

Use the steps in [Exporting Posture Validation Attribute Definitions, page D-48](#), to create a backup of posture validation attribute definitions. You can also use the exported attribute definition file to double-check the vendor ID, application ID, and attribute ID of the posture validation attribute you want to delete.

To delete posture validation attributes, follow these steps:

---

**Step 1** On the computer running Cisco Secure ACS, open an MS DOS command prompt and change directories to the directory containing CSUtil.exe.

**Step 2** Type:

```
CSUtil.exe -delavp vendor-ID  
application-ID  
attribute-ID
```

For more information about vendor, application, and attribute IDs, see [Posture Validation Attribute Definition File, page D-44](#).

CSUtil.exe prompts you to confirm the attribute deletion.

**Step 3** Examine the confirmation prompt and then do one of the following:

- If you are certain you want to delete the attribute identified by the confirmation prompt, type **Y** and press **Enter**.



---

**Tip** You can use the `-q` option to suppress the confirmation prompt.

---

- If you do not want to delete the attribute identified by the confirmation prompt, type **N**, press **Enter**, and return to [Step 2](#).

CSUtil.exe deletes from its internal database the posture validation attribute you specified. In the following example, CSUtil.exe deleted an attribute with a vendor ID of 9876, an application ID of 1, and an attribute ID of 1.

## Posture Validation Attributes

CSUtil v3.3, Copyright 1997-2004, Cisco Systems Inc

Are you sure you want to delete vendor 9876; application 1; attribute 1? (y/n)

y

Vendor 9876; application 1; attribute 1 was successfully deleted

- Step 4** If you are ready to make the attribute deletion take effect, restart the CSAuth and CSAdmin services.



### Caution

---

While CSAuth is stopped, no users are authenticated.

---

To restart the CSAuth, CSLog, and CSAdmin services, enter the following commands at the command prompt, allowing the computer time to perform each command:

```
net stop csauth
net start csauth
net stop cslog
net start cslog
net stop csadmin
net start csadmin
```

Deleted posture validation attributes no longer are available in Cisco Secure ACS.

---

## Default Posture Validation Attribute Definition File

[Example D-2](#) provides the definitions for the posture validation attributes that we provide with Cisco Secure ACS. Should you need to reset the default attributes to their original definitions, use [Example D-2](#) to create a posture validation attribute definition file. For more information about the format of an attribute definition file, see [Posture Validation Attribute Definition File, page D-44](#).

### *Example D-2 Default Posture Validation Attribute Definitions*

```
[attr#0]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00001
```

```
attribute-name=Application-Posture-Token  
attribute-profile=out  
attribute-type=unsigned integer
```

```
[attr#1]  
vendor-id=9  
vendor-name=Cisco  
application-id=1  
application-name=PA  
attribute-id=00002  
attribute-name=System-Posture-Token  
attribute-profile=out  
attribute-type=unsigned integer
```

```
[attr#2]  
vendor-id=9  
vendor-name=Cisco  
application-id=1  
application-name=PA  
attribute-id=00003  
attribute-name=PA-Name  
attribute-profile=in out  
attribute-type=string
```

```
[attr#3]  
vendor-id=9  
vendor-name=Cisco  
application-id=1  
application-name=PA  
attribute-id=00004  
attribute-name=PA-Version  
attribute-profile=in out  
attribute-type=version
```

```
[attr#4]  
vendor-id=9  
vendor-name=Cisco  
application-id=1  
application-name=PA  
attribute-id=00005  
attribute-name=OS-Type  
attribute-profile=in out  
attribute-type=string
```

```
[attr#5]  
vendor-id=9  
vendor-name=Cisco  
application-id=1
```

## Posture Validation Attributes

```
application-name=PA
attribute-id=00006
attribute-name=OS-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#6]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00007
attribute-name=PA-User-Notification
attribute-profile=out
attribute-type=string
```

```
[attr#7]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00001
attribute-name=Application-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#8]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00002
attribute-name=System-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#9]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00006
attribute-name=ServicePacks
attribute-profile=in
attribute-type=string
```

```
[attr#10]
vendor-id=9
```

```
vendor-name=Cisco  
application-id=2  
application-name=Host  
attribute-id=00007  
attribute-name=HotFixes  
attribute-profile=in  
attribute-type=string
```

```
[attr#11]  
vendor-id=9  
vendor-name=Cisco  
application-id=2  
application-name=Host  
attribute-id=00008  
attribute-name=HostFQDN  
attribute-profile=in  
attribute-type=string
```

```
[attr#12]  
vendor-id=9  
vendor-name=Cisco  
application-id=5  
application-name=HIP  
attribute-id=00001  
attribute-name=Application-Posture-Token  
attribute-profile=out  
attribute-type=unsigned integer
```

```
[attr#13]  
vendor-id=9  
vendor-name=Cisco  
application-id=5  
application-name=HIP  
attribute-id=00002  
attribute-name=System-Posture-Token  
attribute-profile=out  
attribute-type=unsigned integer
```

```
[attr#14]  
vendor-id=9  
vendor-name=Cisco  
application-id=5  
application-name=HIP  
attribute-id=00005  
attribute-name=CSAVersion  
attribute-profile=in  
attribute-type=version
```

```
[attr#15]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=00009
attribute-name=CSAOperationalState
attribute-profile=in
attribute-type=unsigned integer
```

```
[attr#16]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=00011
attribute-name=TimeSinceLastSuccessfulPoll
attribute-profile=in
attribute-type=unsigned integer
```

```
[attr#17]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=32768
attribute-name=CSAMCName
attribute-profile=in
attribute-type=string
```

```
[attr#18]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=32769
attribute-name=CSAStates
attribute-profile=in
attribute-type=string
```

```
[attr#19]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00001
attribute-name=Application-Posture-Token
attribute-profile=out
```

```
attribute-type=unsigned integer
```

```
[attr#20]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00002
attribute-name=System-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#21]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00003
attribute-name=Software-Name
attribute-profile=in out
attribute-type=string
```

```
[attr#22]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00004
attribute-name=Software-ID
attribute-profile=in out
attribute-type=unsigned integer
```

```
[attr#23]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00005
attribute-name=Software-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#24]
vendor-id=393
vendor-name=Symantec
application-id=3
application-name=AV
attribute-id=00006
```

```
attribute-name=Scan-Engine-Version  
attribute-profile=in out  
attribute-type=version
```

```
[attr#25]  
vendor-id=393  
vendor-name=Symantec  
application-id=3  
application-name=AV  
attribute-id=00007  
attribute-name=Dat-Version  
attribute-profile=in out  
attribute-type=version
```

```
[attr#26]  
vendor-id=393  
vendor-name=Symantec  
application-id=3  
application-name=AV  
attribute-id=00008  
attribute-name=Dat-Date  
attribute-profile=in out  
attribute-type=date
```

```
[attr#27]  
vendor-id=393  
vendor-name=Symantec  
application-id=3  
application-name=AV  
attribute-id=00009  
attribute-name=Protection-Enabled  
attribute-profile=in out  
attribute-type=unsigned integer
```

```
[attr#28]  
vendor-id=393  
vendor-name=Symantec  
application-id=3  
application-name=AV  
attribute-id=00010  
attribute-name=Action  
attribute-profile=out  
attribute-type=string
```

```
[attr#29]  
vendor-id=3401  
vendor-name=NAI  
application-id=3
```

```
application-name=AV
attribute-id=00001
attribute-name=Application-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#30]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00002
attribute-name=System-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#31]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00003
attribute-name=Software-Name
attribute-profile=in out
attribute-type=string
```

```
[attr#32]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00004
attribute-name=Software-ID
attribute-profile=in out
attribute-type=unsigned integer
```

```
[attr#33]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00005
attribute-name=Software-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#34]
vendor-id=3401
```

```
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00006
attribute-name=Scan-Engine-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#35]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00007
attribute-name=Dat-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#36]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00008
attribute-name=Dat-Date
attribute-profile=in out
attribute-type=date
```

```
[attr#37]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00009
attribute-name=Protection-Enabled
attribute-profile=in out
attribute-type=unsigned integer
```

```
[attr#38]
vendor-id=3401
vendor-name=NAI
application-id=3
application-name=AV
attribute-id=00010
attribute-name=Action
attribute-profile=out
attribute-type=string
```

```
[attr#39]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00001
attribute-name=Application-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#40]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00002
attribute-name=System-Posture-Token
attribute-profile=out
attribute-type=unsigned integer
```

```
[attr#41]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00003
attribute-name=Software-Name
attribute-profile=in out
attribute-type=string
```

```
[attr#42]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00004
attribute-name=Software-ID
attribute-profile=in out
attribute-type=unsigned integer
```

```
[attr#43]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00005
attribute-name=Software-Version
attribute-profile=in out
```

attribute-type=version

```
[attr#44]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00006
attribute-name=Scan-Engine-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#45]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00007
attribute-name=Dat-Version
attribute-profile=in out
attribute-type=version
```

```
[attr#46]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00008
attribute-name=Dat-Date
attribute-profile=in out
attribute-type=date
```

```
[attr#47]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00009
attribute-name=Protection-Enabled
attribute-profile=in out
attribute-type=unsigned integer
```

```
[attr#48]
vendor-id=6101
vendor-name=Trend
application-id=3
application-name=AV
attribute-id=00010
```

```
attribute-name=Action  
attribute-profile=out  
attribute-type=string
```

```
[attr#49]  
vendor-id=10000  
vendor-name=out  
application-id=1  
application-name=CNAC  
attribute-id=00001  
attribute-name=Application-Posture-Token  
attribute-profile=out  
attribute-type=string
```

```
[attr#50]  
vendor-id=10000  
vendor-name=out  
application-id=1  
application-name=CNAC  
attribute-id=00002  
attribute-name=System-Posture-Token  
attribute-profile=out  
attribute-type=string
```

```
[attr#51]  
vendor-id=10000  
vendor-name=out  
application-id=1  
application-name=CNAC  
attribute-id=00003  
attribute-name=Reason  
attribute-profile=out  
attribute-type=string
```

