CSUtil Database Utility

Note The information in this appendix applies to ACS for Windows.

This appendix details the command-line utility, **CSUtil.exe**, for the Cisco Secure Access Control Server Release 4.1, hereafter referred to as ACS. Among its several functions, your can use **CSUtil.exe** 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 by using the ACS System Backup, ACS System Restore, Database Replication, and RDBMS Synchronization features. For more information on these features, see Chapter 8, "System Configuration: Advanced."

This chapter contains the following topics:

Location of CSUtil.exe and Related Files

When you install ACS in the default location, **CSUtil.exe** is located in:

`C:\Program Files\CiscoSecure ACS v`*x.x*`\bin`

where *x.x* is the version of your ACS software. The **CSUtil.exe** tool is located in the *\bin* subdirectory of your ACS installation directory. Files generated by or accessed by **CSUtil.exe** are also located in the *\bin* directory. If you add other files, such as vendor definitions for the ACS dictionary, be sure to put them in the *\bin* directory.

CSUtil Command Syntax

The syntax for the **CSUtil** command is:

```
csutil [-q] [-b backup_filename] [-e number] [-g group_number] [-i file]
[-d [-p secret_key] dump_filename] [-l filename [-passwd secret_key]] [-n]
[-r all|users|config backup_file ] [-u] [-listUDV] [-addUDV slot filename.ini]
[-delUDV slot] [-dumpUDV database_dump_filename]
[-t] [-filepath full_filepath] [-passwd password] [-machine]
(-a | -g group_number | -u user_name | -f user_list_filepath)
[-addAVP filepath] [-delAVP vendor_id application_id attribute_id] [-dumpAVP filename]
[-delPropHPP attribute_ID property_ID] [-delEntHPP attribute_ID entity_name]
```

Table D-1 shows the options that you can use with the **CSUtil** command.

Table D-1 CSUtil Options

Syntax	Use to ...
-q	Use Quiet mode. Does not prompt, use before other options.
-**b** *backup_filename*	Create a system backup.
-**d** [**-p** *secret_key*] *dump_filename*	Dump users and groups database to *dump.txt* or a named file. You should provide a secret key to encrypt user passwords in the dump file.
-e *number*	Decode error number to ASCII message.
-g *group_number*	Dump group information only to *group.txt*.
-i *file*	Import users or NASs from *import.txt* or named file.
-p *secret_key*	Reset password-aging counters during users' and groups' database dump (**-d**).
-l *filename* [**-passwd** *secret_key*]	Empty the user table, initialize profiles, and load users and groups database from *dump.txt* or named file. If you used an encrypt key when dumping the information, you must provide a key to decrypt user passwords and other sensitive information in the dump file.
-n	Empty the user table and shared profile components table, initialize user, group, and network access profiles, and create a new database.
-r all|users|config *backup_file*	Restore a system backup.
-u	List users by group to *users.txt*.
-listUDV	List currently installed user defined vendors (UDVs).
-addUDV*slot filename.ini*	Install user-defined vendor or vendor-specific-attribute (VSA) data from the .ini file.

Table D-1 CSUtil Options (continued)

-delUDV *slot*	Remove a vendor or VSA.
-dumpUDV *database_dump_file*	Dump currently installed vendors to the System UDVs folder.
-t -filepath *full_filepath* **-passwd** *password -machine* (-a | -g *group_number* | -u *user_name* | -f *user_list_filepath*)	Generate protected access credentials (PAC) files for use with Extensible Authentication Protocol-Flexible Authentication via Secure Tunnelling (EAP-FAST) clients. You can generate a user PAC or a machine PAC.
-addAVP *filename*	Add attributes from <*filename*>.
-delAVP *vendor_id application_id attribute_id*	Remove an AVP attribute
-dumpAVP ***filename***	Dump AVP attributes into <*filename*>
-delPropHPP *attribute_ID property_ID*	Remove specific Property from an extended attribute under `Cisco:Host`.
-delEntHPP *attribute_ID entity_name*	Remove specific Entity from an extended attribute under `Cisco:Host`.

Caution Most **CSUtil** options require that you stop the **CSAuth** service. While the **CSAuth** service is stopped, ACS does not authenticate users. To determine if an option requires that you stop **CSAuth**, refer to the detailed topics about the option.

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**, which must be used in conjunction with other options.

Experienced **CSUtil.exe** users might 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 ACS internal data:

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

Backing Up ACS with CSUtil.exe

You can use the **-b** option to create a system backup of all ACS internal data. The resulting backup file has the same data as the backup files that are produced by the ACS Backup feature found in the web interface. For more information about the ACS Backup feature, see ACS Backup, page 7-7.

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

To back up ACS with **CSUtil.exe**:

Step 1 On the computer that is running 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 ACS services during the backup, type **Y** and press **Enter**.

CSUtil.exe generates a complete backup of all 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 ACS that are empty, such as when no administrator accounts exist. These messages apply only to components that are empty, not to the overall success or failure of the backup.

Restoring ACS with CSUtil.exe

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

ACS backup files contain:

- User and group data.
- System configuration.

You can restore user and group data, or system configuration, or both. For more information about the ACS Backup feature, see ACS Backup, page 7-7.

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

To restore ACS with **CSUtil.exe**:

Step 1 On the computer running 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 ACS services during the restoration, type **Y** and press **Enter**.

CSUtil.exe restores the specified portions of your 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 ACS at the time the backup was created.

Initializing the ACS Internal Database

You can use the **-n** option to initialize the ACS internal database. The **-n** option empties the user table and shared profile components table, and initializes user, group, and network access profiles.

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 ACS internal database. Unless you have a current backup or dump of your ACS internal database, all user accounts are lost when you use this option.

To create an ACS internal database:

Step 1 If you have not performed a backup or dump of the ACS internal database, do so now before proceeding. For more information about backing up the database, see Backing Up ACS with CSUtil.exe, page D-3.

Step 2 On the computer that is running 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

Press **Enter**.

The **CSAuth** service stops.

Step 4 Type:

```
CSUtil.exe -n
```

Press **Enter**.

CSUtil.exe displays a confirmation prompt.

Step 5 To confirm that you want to initialize the ACS internal database, type **Y** and press **Enter**.

The ACS internal database is initialized. This process may take a few minutes.

Step 6 To resume user authentication, type:

net start csauth

Press **Enter**.

Creating an ACS Internal Database Dump File

You can use the **-d** option to dump all contents of the ACS internal database into a password-protected text file. You can provide a name for the file; otherwise, it is called *dump.txt*. The dump file provides a thorough and compressible backup of all ACS internal data.

Using the **-l** option, you can reload the ACS internal data from a dump file created by the **-d** option. For more information about the **-l** option, see Loading the ACS Internal Database from a Dump File, page D-7.

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

To dump all ACS internal data into a text file:

Step 1 On the computer that is running 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

Press **Enter**.

The **CSAuth** service stops.

Step 3 To dump to the default *dump.txt* file, type:

```
CSUtil.exe -d -p secret_key
```

Press **Enter**.

CSUtil.exe displays a confirmation prompt.

Step 4 To dump to a named file, type:

```
CSUtil.exe -d -p secret_key dump_filename
```

Press **Enter**.

CSUtil.exe displays a confirmation prompt.

Step 5 To confirm that you want to dump all ACS internal data into a text file, type **Y** and press **Enter**.

CSUtil.exe creates the dump text file. This process may take a few minutes.

Step 6 To resume user authentication, type:

net start csauth

Press **Enter**.

Loading the ACS Internal Database from a Dump File

You can use the **-l** option to overwrite all ACS internal data from a dump text file. This option replaces the existing all ACS internal data with the data in the dump text file. In effect, the **-l** option initializes all ACS internal data before loading it from the dump text file. Dump text files are created by using the **-d** option. You must use the same password used to encrypt the dump files.

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 ACS internal data from a text file:

Step 1 On the computer that is running 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

Press **Enter**.

The **CSAuth** service stops.

Step 3 To load from the default *dump.txt* file, type:

```
CSUtil.exe -l -passwd secret_key
```

where *secret_key* is the same password that was used to encrypt the dump text file. Press **Enter**.

Step 4 To load from a named dump file and reset password-aging counters, type:

```
CSUtil.exe -p -l filename -passwd secret_key
```

where *filename* is the name of the dump file that you want **CSUtil.exe** to use to load ACS internal data. *secret_key* is the same password that was used to encrypt the `dump.txt` file.

Note You must enter `-p` before `-l` as shown in the command line example; otherwise, this operation will not work.

Press **Enter**.

CSUtil.exe displays a confirmation prompt for overwriting all 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 5 To confirm that you want to replace all ACS internal data, type **Y** and press **Enter**.

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

Step 6 To resume user authentication, type:

net start csauth

Press **Enter**.

Cleaning up the ACS Internal Database

Like many relational databases, the ACS internal database marks deleted records as deleted; but does not remove the records from the database. You can clean up the ACS internal database and remove all records marked for deletion by using the following **CSUtil.exe** options:

- **-d**—Export all ACS internal data to a text file, named *dump.txt*.
- **-n**—Create an ACS internal database and index.
- **-l**—Load all ACS internal data from the *dump.txt* file.

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. This process does not necessarily reduce the size of the database.

Note Cleaning up the ACS internal database requires that you stop the **CSAuth** service. While **CSAuth** is stopped, no users are authenticated.

To clean up the ACS internal database:

Step 1 On the computer that is running 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

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** 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 Initializing the ACS Internal Database, page D-5. For more information about the effects of the **-l** option, see Loading the ACS Internal Database from a Dump File, page D-7.

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

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

Step 5 To resume user authentication, type:

net start csauth

Press **Enter**.

User and AAA Client Import Option

You can use the **-i** option to update 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

To import user or AAA client information:

Step 1 If you have not performed a backup or dump of ACS, do so now before proceeding. For more information about backing up the database, see Backing Up ACS with CSUtil.exe, page D-3.

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-10.

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 that is running 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 ACS. Press **Enter**.

CSUtil.exe displays a confirmation prompt for updating the database.

Step 6 To confirm that you want to update ACS with the information from the import text file specified, type **Y** and press **Enter**.

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 must restart CSTacacs and CSRadius for these changes to take effect.

Step 7 To restart CSRadius:

a. Type:

net stop csradius

Press **Enter**. The CSRadius service stops.

b. To start CSRadius, type:

net start csradius

Press **Enter**.

Step 8 To restart CSTacacs:

a. Type:

net stop cstacacs

Press **Enter**. The CSTacacs service stops.

b. To start CSTacacs, type:

net start cstacacs

Press **Enter**.

User and AAA Client Import File Format

This section contains the following topics:

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-2.

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.

Note There are no password character limitations in the ACS user interface, or when using the **CSUtil.exe** to import passwords.

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-2.

Table D-2 ONLINE/OFFLINE Statement Tokens

Token	Required	Value Required	Description
ONLINE	**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 ACS continues to authenticate users during the import.
OFFLINE	**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 ACS. Table D-3 lists the valid tokens for **ADD** statements.

Note **CSUtil.exe** provides no means to specify a particular instance of an external user database type. If an external user database authenticates a user and 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 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 ACS first.

Table D-3 ADD Statement Tokens

Token	Required	Value Required	Description
ADD	Yes	username	Add user information to 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 Challenge Authentication Handshake Protocol (CHAP) password for authentication.
CSDB	No	password	Authenticate the username with the ACS internal database.
CSDB_UNIX	No	UNIX-encrypted password	Authenticate the username with the ACS internal 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 Network Directory Services (NDS) external user database.
EXT_SDI	No	—	Authenticate the username with an RSA external user database.
EXT_ODBC	No	—	Authenticate the username with an Open Database Connectivity (ODBC) external user database.
EXT_LDAP	No	—	Authenticate the username with a generic Lightweight Directory Access Protocol (LDAP) external user database.
EXT_LEAP	No	—	Authenticate the username with a Lightweight and Efficient Application Protocol (LEAP) proxy Remote Access Dial-In User Service (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 ACS internal 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 that a user is assigned to or to update which database ACS uses to authenticate the user.

Table D-4 lists the valid tokens for UPDATE statements.

Table D-4 UPDATE Statement Tokens

Token	Required	Value Required	Description
UPDATE	Yes	username	Update user information to 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. **Note** 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 ACS internal database.
CSDB_UNIX	No	UNIX-encrypted password	Authenticate the username with the ACS internal database by 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:3Al3qf9:CHAP:goodoldchap
```

DELETE Statements

DELETE statements are optional. The DELETE token and its value are required to delete a user account from ACS. The DELETE token, detailed in Table D-5, is the only token in a DELETE statement.

Table D-5 UPDATE Statement Tokens

Token	Required	Value Required	Description
DELETE	Yes	username	The name of the user account to delete.

For example, the following DELETE statement causes **CSUtil.exe** to permanently remove the account with username *John* from the ACS internal 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 ACS.

Table D-6 lists the valid tokens for ADD_NAS statements.

Table D-6 ADD_NAS Statement Tokens

Token	Required	Value Required	Description
ADD_NAS	Yes	AAA client name	The name of the AAA client to add.
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	The authentication protocol that the AAA client uses. For RADIUS, this includes the VSA. **Note** The following values are valid. Quotation marks (“”) are required, due to the spaces in the protocol names. • “TACACS+ (Cisco IOS)” • “RADIUS (Cisco Aironet)” • “RADIUS (Cisco Airespace)” • “RADIUS (Cisco BBSM)” • “RADIUS (Cisco IOS/PIX 6.x)” • “RADIUS (Cisco VPN 3000/ASA/PIX 7.x+)” • “RADIUS (Cisco VPN 5000)” • “RADIUS (IETF)” • “RADIUS (Ascend)” • “RADIUS (Juniper)” • “RADIUS (Nortel)” • “RADIUS (iPass)”
NDG	No	NDG name	The name of the Network Device Group to which to add the AAA client.
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 Adding AAA Clients, page 3-11.
KEEPALIVE	No	Y or N	For AAA clients that are using TACACS+ only, the value set for this token specifies whether the Log Update or Watchdog Packets from this Access Server option is enabled. For more information, see Adding AAA Clients, page 3-11.

For example, the following **ADD_NAS** statement causes **CSUtil.exe** to add the 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-7, is the only token in a **DEL_NAS** statement. **DEL_NAS** statements delete AAA client definitions from ACS.

Table D-7 DEL_NAS Statement Tokens

Token	Required	Value Required	Description
DEL_NAS	Yes	AAA client name	The name of the AAA client to delete.

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

An example of the import text file is:

```
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:vanessaspassword
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 ACS internal 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 ACS internal 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 **CSAuth** service. While **CSAuth** is stopped, no users are authenticated.

To export user information from the ACS internal database into a text file:

Step 1 On the computer that is running 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

Press **Enter**.

The **CSAuth** service stops.

Step 3 Type:

```
CSUtil.exe -u
```

Press **Enter**.

CSUtil.exe exports information for all users in the ACS internal database to a file named *users.txt*.

Step 4 To resume user authentication, type:

```
net start csauth
```

Press **Enter**.

Exporting Group Information to a Text File

You can use the **-g** option to export group configuration data, including shared profile components, from the ACS internal 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 ACS internal database to a text file:

Step 1 On the computer that is running 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
```

Press **Enter**.

The **CSAuth** service stops.

Step 3 Type:

```
CSUtil.exe -g
```

Press **Enter**.

CSUtil.exe exports information for all groups in the ACS internal database to a file named *groups.txt*.

Step 4 To resume user authentication, type:

net start csauth

Press **Enter**.

Decoding Error Numbers

You can use the **-e** option to decode error numbers in ACS service logs. These error codes are internal to ACS. For example, the CSRadius log could contain a message similar to:

```
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 ACS internal error codes only; not to Windows error codes that are sometimes captured in ACS logs, such as when Windows authentication fails.

For more information about ACS service logs, see Service Logs, page 10-12.

To decode an error number from an ACS service log:

Step 1 On the computer that is running 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 in the ACS service log.

Press **Enter**.

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

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

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

In addition to supporting a set of predefined RADIUS vendors and vendor-specific attributes (VSAs), 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 by using RDBMS Synchronization. Custom RADIUS vendor and VSA configurations that you create by using 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 RDMBS Synchronization, see RDBMS Synchronization, page 8-17.

Vendors that you add must be Internet Engineering Task Force (IETF)-compliant; therefore, all VSAs that you add must be subattributes of IETF RADIUS attribute number 26. You can define up to ten custom RADIUS vendors, numbered zero (0) through 9. **CSUtil.exe** allows only one instance of any given vendor, as defined by the unique vendor IETF ID number and 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 ACSs, including the RADIUS vendor slots that the user-defined RADIUS vendors occupy. For more information about database replication, see ACS Internal Database Replication, page 8-1.

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 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 ACS, all 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-21.
- 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-20.

To add a custom RADIUS VSA to ACS:

Step 1 On the computer that is running 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 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, use the command:

```
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 ACS services during the process, type **Y** and press **Enter**.

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

Note While **CSUtil.exe** deletes a custom RADIUS vendor from ACS, all 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 ACS web interface, no AAA client uses the RADIUS vendor. For more information about configuring AAA clients, see Configuring AAA Clients, page 3-8.

Verify that your RADIUS accounting log does not contain attributes from the RADIUS vendor that you want to delete. For more information about configuring your RADIUS accounting log, see Configuring ACS Logs, page 10-13.

To delete a custom RADIUS vendor and VSA set from ACS:

Step 1 On the computer that is running 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-20.

CSUtil.exe displays a confirmation prompt.

Step 3 To confirm that you want to halt all 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 ACS services, deletes the specified RADIUS vendor from ACS. This process may take a few minutes. After it is complete, **CSUtil.exe** restarts ACS services.

Listing Custom RADIUS Vendors

You can use the **-listUDV** option to determine what custom RADIUS vendors are defined in ACS. You also use this option 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 that are defined in ACS:

Step 1 On the computer that is running 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 that this option creates 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 that was used to import the set.

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

ACS places all exported vendor/VSA files in a subdirectory of the directory containing **CSUtil.exe**. The subdirectory is named `System UDVs`. 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 that the current custom RADIUS vendor currently occupies and VSA set. For example, if vendor Widget occupies slot 4, the exported file that **CSUtil.exe** creates is *UDV_4.ini*.

To export custom RADIUS vendor and VSA sets to files:

Step 1 On the computer that is running 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 that is currently configured in ACS, **CSUtil.exe** writes a file in the *\System UDVs* subdirectory.

RADIUS Vendor/VSA Import File

To import a custom RADIUS vendor and VSA set into 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

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-8. 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-8 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 Vendor and VSA Set Definition, page D-22.
Attribute definition	Yes	1 to 255	Defines a single attribute of the VSA set. For more information, see Attribute Definition, page D-22.
Enumeration	No	0 to 255	Defines enumerations for attributes with integer data types. For more information, see Enumeration Definition, page D-23.

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-9 lists valid keys for the vendor and VSA set section.

Table D-9 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 *n* (where *n* is the VSA number)	Yes—you can define 1 to 255 VSAs	Attribute name	The name of a VSA. For each VSA named here, the file must contain a corresponding attribute definition section. **Note** Attribute names must be unique within the RADIUS vendor/VSA import file, and within the set of all RADIUS attributes in ACS. To facilitate unique names, 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 naming convention also makes accounting logs easier to understand.

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 that is defined in the vendor and VSA set section. The section header of each attribute definition section must match the attribute name that is defined for that attribute in the vendor and VSA set section. Table D-9 lists the valid keys for an attribute-definition section.

Table D-10 Attribute Definition Keys

Keys	Required	Value Required	Description
Type	Yes	See description	The data type of the attribute. It must be one of: • STRING • INTEGER • IPADDR If the attribute is an integer, the **Enums** key is valid.

Profile	Yes	See description	The attribute profile defines if the attribute is used for authorization or accounting, or both. The **Profile** key definition must contain at least one of these values:
			• **IN**—The attribute is used for accounting. After you add the attribute to ACS, you can configure your RADIUS accounting log to record the new attribute. For more information about RADIUS accounting logs, see AAA-Related Logs, page 10-1.
			• **OUT**—The attribute is used for authorization.
			In addition, you can use the value `MULTI` to allow several instances of the attribute per RADIUS message.
			Combinations are valid. For example:
			`Profile=MULTI OUT`
			or
			`Profile=IN OUT`

Enums	No (only valid when the TYPE value is INTEGER)	Enumera-tions section name	The name of the enumeration section. **Note** Several attributes can reference the same enumeration section. For more information, see Enumeration Definition, page D-23.

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

You can use enumeration definitions 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 ACS web interface, the text values that you define appear in lists that are 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 can reference an enumeration definition section.

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

Table D-11 lists the valid keys for an enumeration definition section.

Table D-11 Enumerations Definition Keys

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

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 following example RADIUS vendor/VSA import file 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. You can generate PACs for users or for machines. For more information about PACs and EAP-FAST, see EAP-FAST Authentication, page 9-8.

This section contains the following topics:

PAC File Options and Examples

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

- **-filepath** *full_filepath*—Specifies the location of the generated files.
- **-machine**—Use this option to generate PACs for machines instead of users.
- **User specification options**—You must choose one of the four options for specifying the users for whom you want PAC files; otherwise, **CSUtil.exe** displays an error message because no users are specified. User specification options are:
  - **-a**—**CSUtil.exe** generates a PAC file for each user in the ACS internal database. For example, if you have 3278 users in the ACS internal 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 the variable (*N*). ACS has 500 groups, numbered from zero (0) 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 the variable (*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 by using the format *DOMAIN\username*. For example, if you specify `ENGINEERING\augustin`, ACS generates a PAC file named *ENGINEERING_augustin.pac*.

- **-f** *list*—**CSUtil.exe** generates a PAC file for each username in the file that is 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
  echamberlain.pac.
  ```

Tip You can also specify domain-qualified usernames by using the format *DOMAIN\username*. For example, if you specify `ENIGINEERING\augustin`, ACS generates a PAC file named *ENGINEERING_augustin.pac*.

- **-passwd** *password*—**CSUtil.exe** uses the password specified, rather than the default password, to protect the PAC files that it generates. The password that 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 that you devise, rather than the default password.

  PAC passwords can contain any characters and are case-sensitive. They must contain between four and 128 characters. While **CSUtil.exe** does not enforce strong password rules, we recommend that you use a strong password.

  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 9-11.

To generate PAC files:

Step 1 Use the discussion in PAC File Options and Examples, page D-25, to determine the following:

- Which users for whom you want to generate PAC files. If you want to use a list of users, create it now.
- What password to use to protect the PAC files that you generate. If necessary, create a password. Your PAC password should:
  - Be very long.
  - Contain uppercase and lowercase letters.
  - Contain numbers in addition to letters.
  - Contain no common words or names.
- The full path to the directory in which you want the PAC files. If necessary, create the directory.

Step 2 On the computer that is running ACS, open an MS-DOS command prompt and change directories to the directory containing **CSUtil.exe**.

Step 3 To create a PAC file for a user, type:

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

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

Press **Enter**.

To create a PAC file for a machine, type:

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

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

Press **Enter**.

CSUtil.exe generates the PAC files for each user that is 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 to the location that 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 13, "Posture Validation."

This section contains the following topics:

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 comprises a definition header and several of the following described values. For an example of the contents of a posture-validation attribute definition file, see Default Posture-Validation Attribute Definition File, page D-35.

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

name=value

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

Tip Use a semicolon (;) 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 that is required in an attribute definition:

- **[attr#*n*]**—Attribute definition header, where *n* is a unique, sequential integer, beginning with zero (0). **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, in a file with 10 attribute definitions, the fifth attribute is defined as `[attr#5]` instead of `[attr#4]`, **CSUtil.exe** ignores the attribute that is defined as `[attr#5]` and the remaining five attributes that follow 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`, or `attribute-id`. Neither does it limit or define the number of posture-validation attributes that ACS supports.

- **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 that is 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 that are associated with them, identified by the `application-id` value.

- **vendor-name**—A string, the `vendor-name` appears in the ACS web interface and logs for the associated posture-validation attribute. For example, any attribute definition with a `vendor-name` of `9` could have the 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 `application-id` of `1`, which is the Cisco Trust Agent, also known as a posture agent (PA).

- **application-name**—A string, the `application-name` appears in the ACS web 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 the Cisco Trust Agent.

  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 `attribute-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 ACS web 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 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**—ACS accepts the attribute in posture-validation requests and can log the attribute, and you can use it in internal policy rule definitions. Attributes with an `in attribute-profile` are also known as inbound attributes.
  - **out**—ACS can send the attribute in posture-validation responses but you cannot use it in internal policy rule definitions. Attributes with an `out attribute-profile` are also known as outbound attributes. The only outbound attributes that you can configure 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**—ACS 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 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 that are available for defining internal policy rules that use the attribute. An example of an inbound attribute is the `ServicePacks` attribute that the Cisco Trust Agent sends. An example of an outbound attribute is the `System-Posture-Token` attribute, which is sent to the Cisco Trust Agent.

  Valid values of `attribute-type` are:
  - `boolean`
  - `string`
  - `integer`
  - `unsigned integer`
  - `ipaddr`
  - `date`
  - `version`
  - `octet-array`

  For more information about attribute data types, see Posture Validation Attribute Data Types, page 13-6.

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-28. For an example of an attribute-definition file, see Default Posture-Validation Attribute Definition File, page D-35.

To export posture-validation attributes:

Step 1 On the computer that is running 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 posture-validation attribute definitions into ACS 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-28. For an example of an attribute definition file, see Default Posture-Validation Attribute Definition File, page D-35.

Before You Begin

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

Use the steps in Exporting Posture-Validation Attribute Definitions, page D-31, 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:

Step 1 Use the discussion in Posture-Validation Attribute Definition File, page D-28, to create a properly formatted attribute definition file. Place the file in the directory containing **CSUtil.exe** or a directory that is accessible from the computer that is running ACS.

Step 2 On the computer that is running 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 that are specified in the file. An example of a successful addition of nine posture-validation attributes is:

```
C:.../Utils 21: csutil -addavp myavp.txt
...
Attribute 9876:1:11 (Calliope) added to dictionary
Attribute 9876:1:3 (Clio) added to dictionary
Attribute 9876:1:4 (Erato) added to dictionary
Attribute 9876:1:5 (Euterpe) added to dictionary
Attribute 9876:1:6 (Melpomene) added to dictionary
Attribute 9876:1:7 (Polyhymnia) added to dictionary
Attribute 9876:1:8 (Terpsichore) added to dictionary
Attribute 9876:1:9 (Thalia) added to dictionary
Attribute 9876:1:10 (Urania) added to dictionary

AVPs from 'myavp.txt' were successfully added
```

Step 4 If you are ready for the imported attribute definitions to 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
```

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

Importing External Audit Posture-Validation Servers

To create an audit vendor file to import into the ACS dictionary:

Step 1 On the computer that is running ACS, open an MS-DOS command prompt and change to the *\bin* directory (the directory containing **CSUtil.exe**).

Step 2 Type:

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

where *filename* is the name of the file that contains the audit server vendor data. If the file is not located in the *\bin* directory, you must add the full path name.

The format of the file should be:

```
[attr#0]
    vendor-id=<the vendor identifier number>
    vendor-name=<the name of the vendor>
    application-id=6
    application-name=Audit
```

Step 3 Press **Enter**.

Step 4 Restart the CSAdmin CSAuth, and CSLog services. You can restart these services manually from the command prompt, or choose Windows **Programs> Administrative Tools > Services**.

Deleting a Posture-Validation Attribute Definition

The **-delAVP** option deletes a single posture-validation attribute from 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 ACS services is low.

Use the steps in Exporting Posture-Validation Attribute Definitions, page D-31, 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:

Step 1 On the computer that is running 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-28.

CSUtil.exe prompts you to confirm the attribute deletion.

Step 3 Examine the confirmation prompt and then:

- If you are certain that 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 that the confirmation prompt identifies, type **N**, press **Enter**, and return to Step 2.

CSUtil.exe deletes the posture-validation attribute that you specified from its internal database. 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.

```
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 For the attribute deletion to 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 are no longer are available in ACS.

Deleting an Extended Posture-Validation Attribute Definition

To delete the extended posture-validation Property attribute contained in the Cisco:Host application:

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

Step 2 Type:

```
CSUtil.exe -delPropHPP <attribute ID> <property ID>
```

This command removes the specific **PROPERTY** from an Extended attribute under **Cisco:Host**.

For more information about vendor, application, and attribute IDs, see Posture-Validation Attribute Definition File, page D-28.

To delete extended posture-validation **ENTITY** attributes in the **Cisco:Host** application:

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

Step 2 Type:

```
CSUtil.exe -delEntHPP <attribute ID> <entity name>
```

This command removes the specific **ENTITY** from an Extended attribute under **Cisco:Host**.

For more information about vendor, application, and attribute IDs, see Posture-Validation Attribute Definition File, page D-28.

Note Extended attributes are supported only as descendants of the **Cisco:Host** application.

Default Posture-Validation Attribute Definition File

Example D-2 provides the definitions for the posture-validation attributes that we provide with ACS. This example is contained in the file *acs4.0_avp.txt*, in the *\Utils* folder. If you need to reset the default attributes to their original definitions, use the syntax in 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-28.

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-Assessment
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-Assessment
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
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=1
application-name=PA
attribute-id=00008
attribute-name=OS-Release
attribute-profile=in out
attribute-type=string

[attr#8]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00009
attribute-name=Kernel-Version
attribute-profile=in out
attribute-type=version

[attr#9]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00010
attribute-name=Action
attribute-profile=out
attribute-type=string
```

```
[attr#10]
vendor-id=9
vendor-name=Cisco
application-id=1
application-name=PA
attribute-id=00011
attribute-name=Machine-Posture-State
attribute-profile=in out
attribute-type=unsigned integer

[attr#11]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00001
attribute-name=Application-Posture-Assessment
attribute-profile=out
attribute-type=unsigned integer

[attr#12]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00002
attribute-name=System-Posture-Assessment
attribute-profile=out
attribute-type=unsigned integer

[attr#13]
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#14]
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#15]
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#16]
vendor-id=9
vendor-name=Cisco
```

```
application-id=2
application-name=Host
attribute-id=00100
attribute-name=Package
attribute-profile=in
attribute-type=string

[attr#17 (extended)]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00100
attribute-name=Package
entities-list=acrobat;cpio;cups;curl;cvs;cyrus-sasl;emacs;enscript;ethereal;evolution;gaim
;gd;gdk-pixbuf;glibc;gnome-vfs2;gnupg;gtk2;httpd;ia32el;imagemagick;imap;imlib;iproute;ips
ec-tools;kdegraphics;kdelibs;kdenetwork;kdepim;kernel;krb5;less;lftp;lha;libpng;libtiff;li
bxml;libxml2;mailman;mod_python;mozilla;mutt;mysql;mysql-server;nasm;net-snmp;netpbm;nfs-u
tils;openmotif;openoffice.org;openssh;openssl;perl;perl-dbi;php;postgresql;pwlib;python;qt
;realplayer;redhat-config-nfs;rh-postgresql;rsh;rsync;ruby;samba;sharutils;slocate;sox;spa
massassin;squid;squirrelmail;sysstat;tcpdump;telnet;tetex;utempter;vim;xchat;xemacs;xfree8
6;xloadimage;xpdf;zip;
property-id=4
property-name=Version
attribute-profile=in
attribute-type=version

[attr#18 (extended)]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00100
attribute-name=Package
entities-list=acrobat;cpio;cups;curl;cvs;cyrus-sasl;emacs;enscript;ethereal;evolution;gaim
;gd;gdk-pixbuf;glibc;gnome-vfs2;gnupg;gtk2;httpd;ia32el;imagemagick;imap;imlib;iproute;ips
ec-tools;kdegraphics;kdelibs;kdenetwork;kdepim;kernel;krb5;less;lftp;lha;libpng;libtiff;li
bxml;libxml2;mailman;mod_python;mozilla;mutt;mysql;mysql-server;nasm;net-snmp;netpbm;nfs-u
tils;openmotif;openoffice.org;openssh;openssl;perl;perl-dbi;php;postgresql;pwlib;python;qt
;realplayer;redhat-config-nfs;rh-postgresql;rsh;rsync;ruby;samba;sharutils;slocate;sox;spa
massassin;squid;squirrelmail;sysstat;tcpdump;telnet;tetex;utempter;vim;xchat;xemacs;xfree8
6;xloadimage;xpdf;zip;
property-id=5
property-name=Version-String
attribute-profile=in
attribute-type=string

[attr#19]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=00001
attribute-name=Application-Posture-Assessment
attribute-profile=out
attribute-type=unsigned integer

[attr#20]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=00002
attribute-name=System-Posture-Assessment
```

```
attribute-profile=out
attribute-type=unsigned integer

[attr#21]
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#22]
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#23]
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#24]
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#25]
vendor-id=9
vendor-name=Cisco
application-id=5
application-name=HIP
attribute-id=32770
attribute-name=DaysSinceLastSuccessfulPoll
attribute-profile=in
attribute-type=unsigned integer
```

Adding External Audit Device Type Attributes

To create an audit device type attribute file to import into the ACS dictionary:

Step 1 On the computer that is running ACS, open an MS-DOS command prompt and change to the *\bin* directory (the directory containing **CSUtil.exe**).

Step 2 Type:

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

where *filename* is the name of the file that contains the audit server vendor data. If the file is not located in the *\bin* directory, you must add the full path name.

The format of the file should be:

```
[attr#0]
    vendor-id=<the vendor identifier number>
    vendor-name=<the name of the vendor>
    application-id=6
    application-name=Audit
    attribute-id=00012
    attribute-name=Device-Type
    attribute-profile=in out
    attribute-type=string
```

Step 3 Press **Enter**.

Step 4 Restart the CSAdmin CSAuth, and CSLog services. You can restart these services manually from the command prompt, or choose Windows **Programs> Administrative Tools > Services**.